Noname manuscript No.
(will be inserted by the editor)

Formalizing Kant's Rules

A Logic of Conditional Imperatives and Permissives

R. Evans · M. Sergot · A. Stephenson

Forthcoming in the *Journal of Philosophical Logic*

Received: Sept. 2018 / Accepted: Jul. 2019 — pre-revision version, please do not cite

Abstract This paper formalizes part of the cognitive architecture that Kant develops in the *Critique of Pure Reason*. The central Kantian notion that we formalize is the *rule*. A rule, as we interpret Kant, is not a declarative conditional stating what would be true if such and such conditions hold. Rather, a Kantian rule is a general procedure, represented by a conditional impera-tive or permissive, indicating which mental acts must or may be performed. These mental acts are not propositions; they do not have truth-values. Our formalization is related to the input/output logics, a family of logics designed to capture relations between elements that need not have truth-values. In this paper, we introduce KL_3 as a formalization of Kant's conception of rules as conditional imperatives and permissives over mental acts. We explain how it differs from standard input/output logics, geometric logic, and first-order logic, as well as how it translates natural language sentences not well captured by first-order logic. Finally, we show how the various distinctions in Kant's much-maligned Table of Judgements emerge as the most natural way of dividing up the various types and sub-types of rule in KL_3. Our analysis sheds new light on the way in which normative notions play a fundamental role in the conception of logic at the heart of Kant's theoretical philosophy.

Keywords Kant · Rules · Conditional imperatives · Input/output logics · Modality · Normativity

R. Evans
E-mail: RichardPrideauxEvans@imperial.ac.uk

M. Sergot
E-mail: m.sergot@imperial.ac.uk

A. Stephenson
E-mail: andrew.stephenson@soton.ac.uk

1 Introduction

> Judgments, insofar as they are regarded merely as the condition of the unification of given representations in one consciousness, are rules. [*Prolegomena* 4:305][1]

We will define a logic of conditional imperatives and permissives that was designed as part of an effort to make sense of what Kant was trying to do in the *Critique of Pure Reason*. There were two sources of motivation for designing this logic. The first came from our long-term project to extract from the *Critique* a cognitive architecture that could be realised in a computer.

Consider a simple agent with various sensors, trying to make sense of its[2] sensory perturbations. It must, somehow, interpret its motley array of sensory perturbations as representations of an external world. This world consists of objects located in space and persisting through time, causally interacting with each other. What sorts of things must an agent do in order to achieve this? What must an agent do in order to represent a world at all? This is not an epistemological question: we are not asking what conditions have to hold in order for an agent who already believes something to also know something. This is a pre-epistemological question about intentionality: what conditions must hold for an agent to even think a thought that is about the world, irrespective of whether that thought is true or false?

Kant's cardinal innovation, as we read him, is that the agent makes sense of its sensory perturbations by constructing and applying *rules*:

> We have above explained the **understanding** in various ways – through a spontaneity of cognition (in contrast to the receptivity of the sensibility), through a faculty of thinking, or a faculty of concepts, or also of judgements – which explanations, if one looks at them properly, come down to the same thing. Now we can characterize it as the faculty of **rules**. This designation is more fruitful, and comes closer to its essence. Sensibility gives us forms (of intuition), but the understanding gives us rules. It is always busy poring through the appearances with the aim of finding some sort of rule in them. [A126][3]

In making sense of its sensory perturbations, the rules an agent constructs and applies must satisfy various constraints, as yet unspecified. But if they do satisfy these constraints, then the agent achieves what Kant calls "**experience**":

[1] Translations are from the Cambridge Edition of the Works of Immanuel Kant (details at the end), with occasional modifications. With the exception of those to the *Critique of Pure Reason*, which take the standard A/B format, references to Kant are by volume and page number in the Academy Edition [*Immanuel Kants gesammelte Schriften*, 29 volumes, Berlin: de Gruyter, 1902-], along with a short English title.

[2] 'It' will be our default singular third person pronoun: "Through this I, or He, or It (the thing), which thinks, nothing further is represented than a transcendental subject of thoughts = x" [A346/B404].

[3] See also [A52/B76], [A127], [B143], [A132/B171], [A159/B198], [A302/B359], [*Jäsche Logic* 9:11-12], and [*Prolegomena* 4:318]

it has constructed a coherent, unified representation of a coherent, unified external world[4].

If rules are to play this fundamental, load-bearing role in Kant's theory of intentionality – his theory of experience – then we had better be very clear what we mean, exactly, by a rule. If a rule is seen as a conditional that relates *propositions* that have truth-values, then it cannot be a foundational part of his architecture. Kant's project, as we understand it, is to explain intentionality itself: he wants to explain how an agent can have world-directed thoughts that are so much as capable of being true or false. If he presupposes rules that connect propositions that are already true or false, then he has already presupposed too much. We argue that, for Kant, a rule is a general procedure relating *acts*, not propositions. The rule's constituent acts are mental rather than physical. They include things like seeing a bruised apple, feeling a heavy hammer, and hearing a buzzing bee. Crucially, for Kant, such acts do not themselves have truth-values:

> For truth and illusion are not in the object insofar as it is intuited, but in the judgment about it insofar as it is thought. Thus it is correctly said that the senses do not err; yet not because they always judge correctly, but because they do not judge at all. Hence truth, as much as error, and thus also illusion as leading to the latter, are to be found only in judgments, i.e., only in the relation of the object to our understanding... In the senses there is no judgment at all, neither a true nor a false one. [A293-4/B350] See also [*Jäsche Logic* 9:53].

In this paper, we will provide a formalization of Kant's conception of rules as general procedures, using a logic of conditional imperatives and permissives over mental acts. We will also sketch how this logic fits into the larger picture of the Kantian self-legislating agent. We will describe how these conditional imperatives and permissives can be constructed, on the fly, by a rule-induction system that makes sense of its sensory perturbations by spontaneously constructing and applying rules.

We said that there were two sources of motivation for designing this logic. The first was to formalize the notion of a rule at the heart of Kant's cognitive architecture with a view to realising that architecture in a computer. The second source of motivation is exegetical and defensive.

Although Kant's Table of Judgements plays an absolutely pivotal role in his Critical system, it has been roundly criticised. One common objection has been that it is based on the out-dated Aristotelian term logic. Just as Kant's views on nature are based on a defunct conception of Newtonian physics and his views on mathematics are based on a defunct conception of Euclidean geometry, so his views on the mind and its acts of judgement are based on a defunct conception of Aristotelian logic. Yet if the Table of Judgements is incomplete or arbitrary, then the derivation of the Table of Categories and the

[4] Kant uses "experience" ("Erfahrung") in various ways. For this, which we regard as its central use, see especially [Bxli], [A110], [A146/B185], [A225/B272], [A229-30/B282]; [*Prolegomena* 4:292, 320]; [*Metaphysical Foundations of Natural Science* 4:560].

subsequent Transcendental Deduction has failed before it has even started. If the Table of Judgements is based on an incomplete and out-dated logic, then Kant's recurrent use throughout his work of the basic structure it provides is merely the result of an "architectonic mania"[5], rather than the persistent application of a unified template that runs through all our mental activity.

Or so the story goes. Our second motivation for formalizing Kant's conception of rules, then, was to design a logic in which the distinctions he introduces in the Table of Judgements emerge as the most natural way of dividing up the various types and sub-types of rule.

In Section 2, we briefly outline some of the key elements in our preferred interpretation of Kant. We define the essential terms and sketch how they fit together in Kant's theory of experience, focusing on his conception of rules.

In Sections 3, 4, and 5, we present a logic that formalizes Kant's rules as conditional imperatives and permissives over mental acts. We explain how it handles the major deontic paradoxes and how it differs from standard input/output logics, geometric logic, and first-order logic. We also explain how it translates natural language sentences, including those involving multiple quantification and features not captured by first-order logic, such as predicate negation and disjunction.

Finally, in Section 6, we show how this logic makes sense of Kant's Table of Judgements, not only its particular "moments" and its overall structure, but also the various finer points of structure that Kant insists upon.

Our analysis sheds new light on the way in which normative notions play an absolutely fundamental role in the conception of logic at the heart of Kant's theoretical philosophy. Apart from its own intrinsic interest, historical and philosophical, this also provides a clear hint that our analysis might be extended to Kant's account of moral rules and practical agency. Consider the central role that imperatives and permissives play in Kant's moral philosophy (e.g. at [*Groundwork* 4:414ff., 421ff.]), as well as his claim that practical and theoretical reason share a "common principle" [Axx; *Groundwork* 4:391]. We cannot purse this extension here; that is a task for future work.

2 Kant's cognitive architecture

In this section, we outline our preferred interpretation of Kant. The interpretation will be elaborated on and confirmed throughout (especially in Section 6), but we do not attempt to mount a full defence of it here. Our primary aim in this paper is to formalize an aspect of Kant's thought, as we understand it, and then apply the results in making sense of his Table of Judgements. The aspects of our interpretation that we do not defend here have been defended by ourselves or others elsewhere, which we note when relevant[6].

[5] [*Kant and the Capacity to Judge*, p.5].

[6] We are particularly indebted to the writings of Béatrice Longuenesse [38,39], Wayne Waxman [56,57], and Robert Brandom [7–9].

2.1 Mental activity as constituted activity

It is a familiar idea that social activity is *constituted* activity[7]. In certain circumstances, if various constraints are satisfied, then pushing a horse-shaped object across a chequered board *counts as* moving a knight to king's bishop three; an utterance of the words "I do" *counts as* an acceptance of marriage vows; running away *counts as* desertion; writing your name *counts as* signing a contract. Such counts-as claims are constitutive, not merely predicative or classificatory (as when we say that a horse counts as a mammal). We are saying that doing x *just consists in* doing y in the right circumstances; that there is nothing more to doing x than doing y in the right circumstances. In this sense, social actions are things we can only do *mediately*, by doing something else in the right circumstances. The constitution might continue, so that one constituted social activity in turn constitutes another, as when a move in a chess game in turn counts as winning the game. But here, too, we do one thing by doing another, and the circumstances must be appropriate. Neither playing nor winning at chess are things we can do *immediately*.

One action can only count as another if the surrounding circumstances satisfy certain conditions. Just going up to a stranger in the street and saying "I do" does not count as marrying them. Saying "I do" only counts as marrying someone in the particular context of a marriage ceremony when the officiator has asked a particular question. What determines which circumstances are the right circumstances? It is the constitutive *rules* of the constituted activity that determine the subset of circumstances in which doing y also counts as doing x. These *constitutive rules* are to be distinguished from regulative rules, like the driving laws, which merely regulate a pre-existing independent activity.

So we have here a distinction between constituting and constituted activities, where constituted activities can in turn play a role in constituting further activities, as well as an attendant distinction between constitutive and regulative rules. And note finally that, when a constituting activity itself consists in the construction and application of rules, then the constitutive rules that determine when this activity counts as a further, constituted activity will be meta-rules: rules that determine how the construction and application of rules must go if it is to constitute the constituted activity in question.

As we read Kant, the guiding theme of his philosophy of mind is that *mental activity is constituted activity*. His primary concern is with the constituted mental activity **experience**. This is a complex, high-level activity, itself constituted by other constituted mental activities, themselves constituted by yet others, and so on. In each case, we can ask: what constraints must be satisfied for one activity to constitute another? Ultimately, we are asking: what are the constitutive rules of experience? It is the purpose of Kant's cognitive architecture to articulate all of this (and more). He calls it "the conditions for the possibility of experience" [e.g. at A92ff./B124ff., A158ff./B197ff.].

[7] See [47]. For an overview as well as a formal treatment, see [23].

Two of the mental activities that play a role in constituting experience, if certain constraints are satisfied, are **perception** and **judgement**. The former includes things like seeing a bruised apple, feeling a heavy hammer, and hearing a buzzing bee. The latter includes things like forming the thought that all men are mortal, that some men are non-learned, or that Caius is a man. Perception and judgement are distinct but interdependent activities. They are also themselves *constituted* activities, and it is at this level that we come to the four elements of Kant's cognitive architecture that will be essential for our logic.

2.2 The four elements

An **intuition** is a mental object, constructed out of given sensations by the solitary individual agent. Your intuitions are different from my intuitions – each agent has its own private repository. There is no limit to how many intuitions an agent can construct. Intuit*ing* is the constituted mental activity of constructing intuitions[8].

A **mark** is a symbol that can be ascribed to multiple intuitions. Unlike an intuition which can only be had by a single agent, a mark is public and can be used by many different agents. (Marks are general in both of these senses.) A mark has no predefined meaning. Its meaning is determined entirely by its inferential role (i.e. by the rules in which it figures)[9].

A **subsumption** is the mental activity of assigning a mark to an intuition[10] (or tuple of intuitions). As we read Kant, and this is central to everything that follows, a subsumption is an act that does not itself have a truth-value. It is not itself a judgement or a thought, still less a belief or knowledge. Although marks are shared public symbols, intuitions are private mental objects, so the act of subsuming an intuition under a mark is only performable by the particular individual who has that particular intuition.

A **rule** is a general procedure for generating subsumptions from subsumptions. There are two basic types of rule. As Kant describes them:

> the representation of a universal condition in accordance with which a certain manifold (of whatever kind) **can** be posited is called a **rule** [*Regel*], and, if it **must** be so posited, a **law** [*Gesetz*]. [A113][11]

[8] We have argued for this traditional view of intuition, and against the currently popular 'relationalist' view elsewhere. See [50].

[9] See [36].

[10] See [*Kant and the Capacity to Judge*, p.92n] and also [56] p.264 and p.269n.

[11] Kant rarely sticks to this rule/law terminology and we do not adopt it here, referring to both imperatives and permissives simply as rules. See also [B201n], where Kant uses yet other terminology: "All **combination** (*conjunctio*) is either **composition** (*compositio*) or **connection** (*nexus*). The former is the synthesis of a manifold of **what does not necessarily belong to each other**... The second combination (*nexus*) is the synthesis of that which is manifold insofar as they **necessarily** belong to one another".

A rule is not a sentence – it is a general *procedure*. But if it were to be *described* by a sentence, it would be described by a conditional imperative or permissive. For example:

> for every intuition x, if you subsume x under mark p, then also subsume x under mark q!

Or:

> for every intuition x, if you subsume x under mark p, then feel free[12] to also subsume x under mark r!

Rules are general procedures that apply to all intuitions. Unlike subsumptions, which are private to the individual, rules are things that the solitary agent can share with others[13]. Unlike subsumptions, which bring two heterogenous elements together (the intuition and the mark), rules bring homogenous elements (various subsumptions) together.

To reiterate, rules are not *themselves* conditional imperatives or permissives like those above. This is important because natural language conditional imperatives and permissives have truth-evaluable content in their antecedents, whereas this is not the case for Kantian rules (for the reasons given in Section 1). Our claim is that Kantian rules, as general procedures, can nevertheless be formalized using a logic of conditional imperatives and permissives (with a suitable formal semantics). As a convenience we will often talk as though rules just are conditional imperative or permissives, but strictly speaking what we mean is that they can be described or formalized as such.

These, then, are the four basic elements of Kant's constitutive psychology: intuitions, marks, subsumptions, and rules. If certain constraints are satisfied, if everything comes together, then:

- an intuition counts as a *representation of a particular external object*
- a mark counts as a *concept*
- a subsuming counts as a *perception*
- a rule counts as a *judgement* (with a truth-value)

As we read Kant, these constitutive counts-as claims should not be thought of as successive or independent stages. An intuition only counts as a representation of an external particular insofar as it is subsumed under a mark that counts as a concept; a mark only counts as a concept insofar as it is involved in a subsumption that counts as a perception; and a subsumption only counts as a perception insofar as it is bound to other subsumptions in a rule that counts as a judgement; thus, in turn, an intuition only counts as a representation of an external particular and a mark only counts as a concept insofar as

[12] This informal way of putting it is not ideal. What we are trying to express here is the permissive that corresponds to the imperative as "may" corresponds to "must". If English contained a punctuation mark corresponding to "!" that represented a permissive rather than an imperative, then we would use that, but there isn't one.

[13] [*Kant and the Capacity to Judge*, p.88]: "This is how, by virtue of its logical form alone, a judgment lays a claim to holding for *any consciousness*, whereas a mere coordination of representations might only hold for my subjective consciousness.". See also [32] and [51].

each figures in a rule that counts as a judgement. And a rule only counts as a judgement insofar as it is bound to other such rules in a coherent, unified representation of a coherent, unified external world; that is, only insofar as it is part of *experience*.

This, in a nutshell, is Kant's constitutive theory of experience. Its constitutive (meta-) rules – the constraints that must be satisfied for the above counts-as claims to hold – are what Kant articulates in the Analytic of Principles[14]. Its basic elements are intuitions, marks, subsumptions, and rules, all of which will play a role in our logic. Here we focus on just one of the counts-as claims: that a rule counts as a judgement.

2.3 Judgements as rules

Recall that a rule is a general procedure that we will formalize as a conditional imperative or permissive. It might seem strange to think of an imperative or permissive rule as something that can count as having a truth-value, but this, we contend, is exactly what Kant has in mind. He says:

> All judgements are accordingly functions of unity among our representations, since instead of an immediate representation, a higher one, which comprehends this and other representations under itself, is used for the cognition of the object, and many possible cognitions are thereby drawn together into one [A69/B94]

This is exactly the role of rules, on our account. A rule is a "higher representation" that binds together subsumptions, which consist of marks and intuitions, or "immediate representations". It is "the mediate cognition of an object, hence the representation of a representation of it" [A68/B93]. Thus:

> Judgments, insofar as they are regarded merely as the condition of the unification of given representations in one consciousness, are rules. [*Prolegomena* 4:305]

> All rules (judgments) contain objective unity of consciousness of the manifold of cognition, hence a condition under which one cognition belongs with another to one consciousness. [*Jäsche Logic* 9:121]

Consider Kant's identification of judgements with rules. This identification is easiest to see in the case of universal judgements. The judgement "All men are mortal" just is the rule:

> for every intuition x, if you subsume x under "man", then also subsume x under "mortal"!

But the identification applies equally to particular judgements. The difference is that particular judgements are permissive rules. "Some men are non-learned" just is the rule:

[14] How exactly this goes has been interpreted in many different ways. See, for example, [10,33, 34,38,56,58]. For our own preferred interpretation, implemented in the aforementioned computer model, see [17,18].

> for every intuition X, if you subsume X under "man", then feel free to also subsume X under "non-learned"!

And it also applies to singular judgements. "Caius is a man" just is the rule:

> for every intuition x, if you subsume x under "Caius", then also subsume x under "man"!

together with a constraint:

> for any distinct intuitions x and y, do not subsume both x and y under "Caius"!

Indeed, after presenting our logic in Sections 3, 4, and 5, we will argue in Section 6 that our rule-based analysis accounts for all of the different kinds of judgement that Kant identifies in his Table of Judgements, and we will also show how it accounts for the Table's finer points of structure. Note, for instance, how singular judgements come out above as a sub-type of universal judgements; how both singular judgements and universal judgements imply particular judgements, so long as imperatives imply permissives; and how negation can be applied to a predicate within an atomic (categorical) judgement.

2.4 What Kant meant by "logic"

We have become accustomed to thinking of logic as the study of entailment relations between sets of linguistic items. We are given a set of sentences, A, and a further sentence p and we want to find out if A entails p, written $A \models p$, where $\models$ is defined in terms of truth: $A \models p$ if any model in which A is true is also a model in which p is true.

Logic, for Kant, was not primarily about entailment relations between elements with truth-values. First and foremost, it is a description of *how we should think*[15]. This project will turn out to *include* an account of entailment relations between elements with truth-values. But, we contend, it is not *exhausted* by such an account.

Our focus in this paper is on what Kant calls "transcendental" logic: a description of how we should think when our goal is experience[16]. As we read Kant, the fundamental question of transcendental logic is: given a collection of subsumptions that the agent is performing concurrently, and a set of rules it has adopted, what further subsumptions may/must it also perform, if it is to achieve experience? Note that this is a question about mental acts: given that the agent is performing these mental acts, and given that it has adopted these rules, what further mental acts may/must it perform? These mental

[15] See [A52ff./B76ff.], [A131/B170], and [*Jäsche Logic* 9:14].

[16] See [1] for a rather different formalization of transcendental logic. Their inverse systems are used to model the way objects of intuition are constructed over time through the dynamical process of synthesis. Though the details differ, the pioneering work of Achourioti and Lambalgen (see also [2]) has been a significant source of inspiration for the current project.

acts (subsumptions) do not have truth-values. Transcendental logic is not primarily a logic relating elements that have truth-values. Rather it tells us what mental acts we may/must perform.

Suppose, for example, the agent has adopted the following rules:

- If you are seeing yellow and black stripes and hearing a buzzing, then feel free to perceive a bee!
- If you are seeing yellow and black stripes and hearing a buzzing, then feel free to perceive a wasp!
- Do not count anything as both a wasp and a bee!

Having adopted the above rules, suppose that the agent now performs the subsumptions that, in the right circumstances, constitute the antecedents: seeing yellow and black stripes and hearing a buzzing. What further subsumptions may/must it make? To repeat, this is not yet a question about which judgements it should adopt. It is not yet a question about which propositions it should hold for true. Rather, it is a question about what mental acts it should perform. In the case at hand, it is a question about what it should *perceive*[17]. One permissible subsumption would be to perceive a bee. Another permissible subsumption would be to perceive a wasp. But it is not permissible to subsume the same intuition under both "bee" and "wasp". Transcendental logic describes the sets of permissible mental activities available, given a set of rules that have already been adopted.

There is, however, so conceived, a further, secondary question for transcendental logic to answer: given a collection of judgements (i.e. rules) that have been adopted, what other judgements may/must also be adopted? Experience is constituted by perception *and* judgement. Now, since judgements do have truth-values, this secondary aspect of transcendental logic aligns with Frege's focus on truth-conditional logic. In this paper, we present a logic that addresses both aspects of transcendental logic (see Section 3.6). But our logic begins "one level down" from first-order logic[18]: it represents (the perceptual) activities that do not themselves have truth-values but which can contribute to the constitution of (the judgemental) elements that do have truth-values, all of which together, if things go right, will constitute experience.

3 KL_1

In the following three sections, we present a logic that formalizes Kant's rules as conditional imperatives and permissives over mental acts. Our claim is not, of course, that Kant had this precise logic in mind. Rather, the claim is

[17] Note that these mental acts are at the sub-personal level – our 'agent' is a sub-personal rule-induction system. It is not as if the person consciously chooses whether to perceive a bee or a wasp, but rather that a pre-conscious process makes this "decision". It does so with the goal of achieving experience, whence the sense and force of a question about what *should* be perceived. But experience, on Kant's account, is the first level at which there arises anything like a person's conscious perspective on a unified, coherent external world. See [49] for further discussion.

[18] See [1], p.236 for a related claim.

that our formalization is based on, compatible with, and helps explain part of Kant's view in the *Critique of Pure Reason* (and associated texts, especially *Jäsche Logik*).

From what has already been said, we know this logic must satisfy two constraints. First, since subsumptions are *acts*, we need a logic that does not assume its constituent elements have truth-values. The input/output logics [41] were designed to capture inferential relations between elements that do not necessarily have truth-values. They were conceived in response to *Jørgensen's Dilemma* (a trilemma):

1. Logical inference requires that the elements (the premises and conclusions) have truth-values.
2. Imperatives do not have truth-values. For example, the command "Burn all the books!" has a *satisfaction*-condition, but does not have a truth-value.
3. There are valid logical inferences between imperatives. For example:
   - Burn all the books in the library!
   - *The Critique of Pure Reason* is a book in the library
   - Therefore: burn *The Critique of Pure Reason*!

The input/output logics resolve this impasse by denying the first claim: they support inference between elements that do not have truth-values. The logic we present below is a member of the family of input/output logics (broadly conceived).

The second constraint on any logic that formalizes Kant's rules is that it must support not only conditional imperatives but also conditional *permissives*. For example:

> if you subsume intuition x under mark p, then feel free to also subsume x under q!

A logic that contains explicit permissives as well as imperatives will generate *multiple* acceptable sets of derived subsumptions[19].

For ease of exposition, we divide the logic into three parts. The first part, KL_1, is a type of input/output logic with two types of rule: conditional imperatives and permissives. The second part, KL_2, extends KL_1 with a negation operator. The third part, KL_3, extends KL_2 by adding variables and quantifiers.

All three logics, KL_1, KL_2, and KL_3, have been implemented and tested in computer programs. In particular, the soundness, completeness, and monotonicity for KL_1 and KL_2 have been empirically verified[20]. The code corresponds closely to the text in Sections 3, 4, and 5 below.

[19] The standard input/output logics generate a single set of derived conclusions. The family of logics we present here are unusual (in the family of input/output logics) in generating multiple acceptable sets of conclusions. Of course, many non-monotonic formalisms generate multiple acceptable sets of conclusions. See Section 3.7 for further comparison.

[20] See `https://github.com/RichardEvans/kl_haskell`.

Table 1: Example rules in KL_1

Rule	Readable version	Translation
$\{p\} \boxdot\!\!\rightarrow \{q\}$	$p \boxdot\!\!\rightarrow q$	If you are doing p, then do q!
$\{p,q\} \boxdot\!\!\rightarrow \{r\}$	$p \wedge q \boxdot\!\!\rightarrow r$	If you are doing p and q, then do r!
$\{p\} \boxdot\!\!\rightarrow \{q,r\}$	$p \boxdot\!\!\rightarrow q \vee r$	If you are doing p, then do q or do r!
$\{\} \boxdot\!\!\rightarrow \{p\}$	$\top \boxdot\!\!\rightarrow p$	Do p!
$\{p\} \boxdot\!\!\rightarrow \{\}$	$p \boxdot\!\!\rightarrow \bot$	Don't, whatever you do, do p!
$\{p\} \diamond\!\!\rightarrow \{q\}$	$p \diamond\!\!\rightarrow q$	If you are doing p, then feel free to do q!
$\{p,q\} \diamond\!\!\rightarrow \{r\}$	$p \wedge q \diamond\!\!\rightarrow r$	If you are doing p and q, then feel free to do r!
$\{p\} \diamond\!\!\rightarrow \{q,r\}$	$p \diamond\!\!\rightarrow q \vee r$	If you are doing p, then feel free to do q or r!
$\{\} \diamond\!\!\rightarrow \{p\}$	$\top \diamond\!\!\rightarrow p$	Feel free to do p!

3.1 Syntax

Let $\mathcal{A}$ be the set of all atoms. A, B, C and X will range over sets of atoms, and a, b, c will range over individual atoms.

There are two types of rule in KL_1:

$$\mathcal{R} ::= B \boxdot\!\!\rightarrow C \mid B \diamond\!\!\rightarrow C$$

B is the body of the rule; it is a set of atoms representing a conjunction. C is the head of the rule; it is a set of atoms representing a disjunction. We use sets, not sequences or multisets, to avoid various uninteresting inferences involving duplication and permutation of elements.

For readability, we write the body of the rule as a conjunction, and the head of a rule as a disjunction. The rule $\{p,q\} \boxdot\!\!\rightarrow \{r,s\}$ is represented as:

$$p \wedge q \boxdot\!\!\rightarrow r \vee s$$

If the body of the rule is empty, we write $\top$ instead of the empty set. For example, $\{\} \diamond\!\!\rightarrow \{p\}$ is written as $\top \diamond\!\!\rightarrow p$. If the head of the rule is empty, we write $\bot$ for the empty set. For example, $\{p\} \boxdot\!\!\rightarrow \{\}$ is written as $p \boxdot\!\!\rightarrow \bot$. Rules with empty bodies and singleton heads are called **facts**, while rules with empty heads are called **constraint rules**.

The $\boxdot\!\!\rightarrow$ rules are intended to be read as *conditional imperatives between actions*. For example, the rule $p \boxdot\!\!\rightarrow q \vee r$ should be read as "if you are doing p, then do q or do r!".

The $\diamond\!\!\rightarrow$ rules are intended to be read as *conditional permissives between actions*. For example, $p \diamond\!\!\rightarrow q$ should be read as "if you are doing p, then feel free to do q!". Some example rules are given in Table 1.

Since the elements of rules are *actions* – not propositions with truth-values – disjunction should not be interpreted truth-functionally. To say that you must do $p \vee q$ is to say there are two available actions, p and q, and you must choose one of these actions (or both[21]).

[21] Kant's disjunctions are exclusive ([A73-4,B99], [*Jäsche Logic* p.106]), while ours are not. We represent an exclusive disjunction by a non-exclusive disjunction $\top \boxdot\!\!\rightarrow p \vee q$ plus a constraint $p \wedge q \boxdot\!\!\rightarrow \bot$.

It is straightforward to generalise the form of rules to allow disjunctions of conjunctions of atoms in the head. We will not do that here for ease of exposition.

3.2 Semantics (part 1)

Given a (countable but not necessarily finite) set R of rules and a (finite) set $A \subseteq \mathcal{A}$ of atoms, the consequences $out_1(R, A)$ is a set $\{X_1, X_2, \dots\}$ of sets of atoms, where each $X_i \subseteq \mathcal{A}$ is one of the distinct ways in which the rules can be satisfied.

There are two sources of non-determinism in KL_1. The first is disjunction. Rules that have disjunctive heads can be satisfied in multiple ways. For example, given:

$$R = \begin{cases} p \boxdot\!\!\rightarrow q \vee r \\ q \boxdot\!\!\rightarrow s \end{cases}$$

with $A = \{p\}$, the possible outcomes are:

$$out_1(R, A) = \begin{cases} \{p, q, s\} \\ \{p, r\} \\ \{p, q, r, s\} \end{cases}$$

The second source of non-determinism is $\diamond\!\!\rightarrow$ rules. For example, given:

$$R = \begin{cases} \top \diamond\!\!\rightarrow p \\ \top \diamond\!\!\rightarrow q \\ p \wedge q \boxdot\!\!\rightarrow \bot \end{cases}$$

the possible outcomes are:

$$out_1(R, \{\}) = \begin{cases} \{\} \\ \{p\} \\ \{q\} \end{cases}$$

The out_1 function is defined in terms of a set $cns(R, A)$ of consequences, representing the various ways of applying R to A, from which are filtered out those that do not satisfy the rules in R.

Definition 1 A set X of atoms satisfies a set R of rules, written $X \models R$, when X satisfies every rule in R. X satisfies a rule r, written $X \models r$, when:

$$\begin{array}{ll} X \models B \boxdot\!\!\rightarrow C & \text{if } B \not\subseteq X \text{ or } C \cap X \neq \emptyset \\ X \models B \diamond\!\!\rightarrow C & \text{always} \end{array}$$

Definition 2 For all sets R of rules and sets A of atoms:

$$out_1(R, A) = \{X \in cns(R, A) \mid X \models R\}$$

$cns(R, A)$ is defined inductively as follows:

$$\begin{aligned} cns_0(R, A) &= \{A\} \\ cns_{n+1}(R, A) &= \{X \cup \{t\} \mid X \in cns_n(R, A),\ t \in step(R, X)\} \\ cns(R, A) &= \bigcup_{n \geq 0} cns_n(R, A) \end{aligned}$$

The *step* function combines the consequences of the various rules that apply:

$$step(R, X) = \{c \mid B \boxdot\!\!\rightarrow C \in R \text{ or } B \diamond\!\!\rightarrow C \in R,\ B \subseteq X,\ c \in C\}$$

The *step* function treats $\boxdot\!\!\rightarrow$ and $\diamond\!\!\rightarrow$ exactly the same; the place where they are treated differently is in the satisfaction condition $X \models R$ in out_1.

Note that, according to this semantics, the permissives are **weak** in that they are *overridden by the imperatives*. If we have $p \diamond\!\!\rightarrow q$ and $p \wedge q \boxdot\!\!\rightarrow \bot$, then the constraint overrides the $\diamond\!\!\rightarrow$ rule:

$$\begin{aligned} R &= \begin{cases} p \diamond\!\!\rightarrow q \\ p \wedge q \boxdot\!\!\rightarrow \bot \end{cases} \\ A &= \{p\} \\ out_1(R, A) &= \{\{p\}\} \end{aligned}$$

Similarly, if we have $p \boxdot\!\!\rightarrow q$ and $p \diamond\!\!\rightarrow r$ with q and r incompatible (i.e., $q \wedge r \boxdot\!\!\rightarrow \bot$), then $p \boxdot\!\!\rightarrow q$ will trump $p \diamond\!\!\rightarrow r$:

$$\begin{aligned} R &= \begin{cases} p \boxdot\!\!\rightarrow q \\ p \diamond\!\!\rightarrow r \\ q \wedge r \boxdot\!\!\rightarrow \bot \end{cases} \\ A &= \{p\} \\ out_1(R, A) &= \{\{p, q\}\} \end{aligned}$$

It is also worth observing that the assumptions A can be replaced equivalently by a set of corresponding unconditional $\boxdot\!\!\rightarrow$ rules. For any set of assumptions A:

$$out_1(R, A) = out_1(R \cup \{\top \boxdot\!\!\rightarrow a \mid a \in A\}, \emptyset)$$

For readability we sometimes write $out_1(R)$ as shorthand for $out_1(R, \emptyset)$.

Note also that $out_1(R, A)$ is not monotonic in either R or A.

It remains to confirm that *cns* and out_1 are well-defined and unique for any set of rules R and assumptions A. We do this in the next section by translating rules R to a simple form of logic program for which the required properties are immediate.

3.3 Semantics (part 2)

In this section, we provide an alternative semantics that is provably equivalent to the semantics in Section 3.2 above.

A **definite clause** is a rule of the form

$$c \leftarrow b_1, \ldots, b_n \qquad (n \geq 0)$$

where c and $b_1, \ldots, b_n$ are atoms. We are using standard logic programming notation: c is the head of the clause; the body $b_1, \ldots, b_n$ is to be read as a conjunction. As is usual, where the body of a clause is empty we identify a clause $c \leftarrow$ with the the atom c. A definite logic program is a set of definite clauses.

The idea is that every $\Box\!\!\rightarrow$ and $\Diamond\!\!\rightarrow$ rule can be translated to a set of definite clauses, each of which represents one of the ways that the rule can be satisfied; a set of rules is translated to the set of definite programs obtained by taking all combinations of the translations of the individual rules.

Definition 3 (Definite clause encoding) Define a function def_r from rules to sets of sets of definite clauses:

$$def_r(B \Box\!\!\rightarrow C) = \begin{cases} \{\emptyset\} \text{ if } C = \emptyset \\ \{\{c \leftarrow B \mid c \in C'\} \mid C' \subseteq C,\ C' \neq \emptyset\} \text{ otherwise} \end{cases}$$

$$def_r(B \Diamond\!\!\rightarrow C) = \{\{c \leftarrow B \mid c \in C'\} \mid C' \subseteq C\}$$

Now define a function def that translates a set of rules into a set of definite programs (a set of sets of definite clauses):

$$def(\{r_1, r_2, \ldots\}) = \{D_1 \cup D_2 \cup \cdots \mid D_1 \in def_r(r_1), D_2 \in def_r(r_2), \ldots\}$$

Example 1

$$def_r(p \Diamond\!\!\rightarrow q \vee r) = \begin{cases} \{\} \\ \{q \leftarrow p\} \\ \{r \leftarrow p\} \\ \{q \leftarrow p,\ r \leftarrow p\} \end{cases}$$

$$def_r(r \Box\!\!\rightarrow s) = \{\{s \leftarrow r\}$$

$$def(\{p \Diamond\!\!\rightarrow q \vee r,\ r \Box\!\!\rightarrow s\}) = \begin{cases} \{s \leftarrow r\} \\ \{q \leftarrow p, s \leftarrow r\} \\ \{r \leftarrow p, s \leftarrow r\} \\ \{q \leftarrow p,\ r \leftarrow p, s \leftarrow r\} \end{cases}$$

Definition 4 (Least model) Let $T_D \colon 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ be the 'immediate consequence operator' [16] of the definite program D:

$$T_D(X) = \{c \mid c \leftarrow B \in D,\ B \subseteq X\}$$

For any $A \subseteq \mathcal{A}$, let $\mathrm{M}(D, A)$ be defined inductively as follows:

$$M_0(D, A) = A$$
$$M_{i+1}(D, A) = M_i(D, A) \cup T_D(M_i(D, A))$$
$$\mathrm{M}(D, A) = \bigcup_{i \geq 0} M_i(D, A)$$

The following are all properties of definite logic programs [16] found in any standard text on logic programming. $\mathrm{M}(D, A)$ is the least fixpoint of T_D that contains A. For definite clauses D it always exists and is unique. $\mathrm{M}(D, A)$ is also the least (set inclusion) set of atoms containing A and closed under the rules D, and the least (Herbrand) model of $D \cup A$. (We are identifying an atom a with the clause $a \leftarrow$.)

Now we can define an alternative version of *cns* in terms of *def* and M.

$$cns_d(R, A) = \{\mathrm{M}(D, A) \mid D \in def(R)\}$$

In the original semantics of Section 3.2, the inductive definition of *cns* can be seen as the construction of a tree rooted in $\{A\}$ whose leaves are the elements of $cns(R, A)$. Here in our second, alternative semantics, $cns_d(R, A)$ can be seen as a set of linear derivations each of which is the application to A of one of the definite programs in $def(R)$.

Clearly if $D \in def(R)$ then $\mathrm{M}(D, A)$ satisfies every rule of the form $B \ \Box\!\!\rightarrow C$ ($C \neq \emptyset$) in R and (trivially) every $\Diamond\!\!\rightarrow$ rule in R.

It can be confirmed (e.g., by induction on R) that if R contains no constraint rules, i.e., no rules of the form $B \ \Box\!\!\rightarrow \bot$, then:

$$cns_d(R, A) = \{X \in cns(R, A) \mid X \models R\}$$

Let $R_\bot$ denote the set of all constraint rules in R. Then:

$$out_1(R, A) = \{X \in cns_d(R, A) \mid X \models R_\bot\}$$

Putting the above together we have the following alternative characterisation of $out_1(R, A)$.

Proposition 1 *Let R be a set of rules and A a set of atoms.*

$$out_1(R, A) = \{\mathrm{M}(D, A) \mid D \in def(R),\ \mathrm{M}(D, A) \models R\}$$

Proof In the preceding discussion. □

The following corollary will be useful later:

Proposition 2 *If R is a set of rules and X a set of atoms, then*

$$\textit{if } X \models R \textit{ then } X \in out_1(R, X)$$

Proof Assume $X \models R$. We shall provide a definite program D in $def(R)$ such that $M(D, X) = X$. For each rule $r_1, ..., r_n$ in R, construct definite programs $d_1, ..., d_n$ as follows. If $r_i = B \Diamond\!\!\rightarrow C$, then let $d_i = \{\}$. If $r_i = B \ \Box\!\!\rightarrow C$, consider two cases. First, if $B \not\subseteq X$, let $d_i = \{\}$. Second, if $B \subseteq X$, then since $X \models B \ \Box\!\!\rightarrow C$, $X \cap C \neq \emptyset$. Let $C' = X \cap C$ and define d_i as $\{c \leftarrow B \mid c \in C'\}$. Let $D = \{d_1, ..., d_n\}$. We have $M(D, X) = X$ and hence by Proposition 1, $X \in out_1(R, X)$. □

3.4 Entailment

We shall define entailment on KL_1 rules using a notion of *strong equivalence* between rule sets. It is natural to say that two rule sets R_1 and R_2 are strongly equivalent if, for all rule sets R' and all sets A of assumptions:

$$out_1(R_1 \cup R', A) = out_1(R_2 \cup R', A)$$

Since $out_1(R, A) = out_1(R \cup \{\top \boxminus\!\!\rightarrow a \mid a \in A\}, \emptyset)$, it is equivalent to require that $out_1(R_1 \cup R', \emptyset) = out_1(R_2 \cup R', \emptyset)$ for all rule sets R'. However, for strong equivalence of KL_1 rule sets it is sufficient to restrict attention to sets R' of *nullary* rules ('facts'), i.e., rules of the form $\top \boxminus\!\!\rightarrow a$. This is because $out_1(R, \emptyset)$ can be seen as being defined by a set of definite clause programs $def(R)$. Two definite clause programs D_1 and D_2 have the same models, and in particular the same least models, if and only if $D_1 \cup A$ and $D_2 \cup A$ have the same models for all sets A of unit clauses ('facts'), i.e., clauses of the form $a \leftarrow \top$. The property generalises straightforwardly to comparing sets of definite clause programs as required here.

We therefore take the following definition of strong equivalence and of rule entailment.

Definition 5 Two rule sets R_1 and R_2 are **strongly equivalent** in KL_1 if:

$$out_1(R_1, A) = out_1(R_2, A) \quad \text{for all sets } A \text{ of atoms}$$

A set R of rules **entails** a rule r in KL_1, written $R \models_{\mathrm{KL}_1} r$, if R and $R \cup \{r\}$ are strongly equivalent in KL_1. In other words, $R \models_{\mathrm{KL}_1} r$ if

$$out_1(R, A) = out_1(R \cup \{r\}, A) \quad \text{for all sets } A \text{ of atoms}$$

It is also convenient to employ a functional notation. $kl_1(R)$ denotes the set of rules semantically entailed by R in KL_1: $kl_1(R) = \{r \mid R \models_{\mathrm{KL}_1} r\}$. Rule sets R_1 and R_2 are strongly equivalent in KL_1 when $kl_1(R_1) = kl_1(R_2)$.

A set R of rules is **strongly inconsistent** in KL_1 if, for every set A of atoms, $out_1(R, A) = \emptyset$.

Proposition 3 *Let R be a set of rules. $out_1(R, A) = \emptyset$ for all sets A of atoms if and only if $out_1(R, \emptyset) = \emptyset$. Hence, R is strongly inconsistent in KL_1 when $out_1(R, \emptyset) = \emptyset$.*

Proof We prove that if $out_1(R, A) \neq \emptyset$ for some A then $out_1(R, \emptyset) \neq \emptyset$. For suppose $X \in out_1(R, A)$. Then there is some definite program D_R in the encoding $def(R)$ of R such that $X = \mathrm{M}(D_R, A)$ and $X \models R_\perp$ where $R_\perp$ are the constraint rules $B \boxminus\!\!\rightarrow \perp$ of R. But $\mathrm{M}(D_R, \emptyset) \subseteq \mathrm{M}(D_R, A)$ so $\mathrm{M}(D_R, \emptyset) \models R_\perp$, and $\mathrm{M}(D_R, \emptyset) \in out_1(R, \emptyset)$. The other direction is trivial. □

Notice that if $out_1(R, A) = \emptyset$ for all A then, for all R', $out_1(R \cup R', A) = \emptyset$ for all A, and hence $R \models_{\mathrm{KL}_1} R'$. Now suppose $R \models_{\mathrm{KL}_1} \top \boxminus\!\!\rightarrow \perp$. Then $out_1(R, A) = out_1(R \cup \{\top \boxminus\!\!\rightarrow \perp\}, A)$ for all A, and clearly $out_1(R \cup \{\top \boxminus\!\!\rightarrow \perp\}, A) = \emptyset$ because $X \not\models \top \boxminus\!\!\rightarrow \perp$ for any X. So we have the following.

Proposition 4 *Let R be a set of rules. R is strongly inconsistent in KL_1 iff $R \models_{KL_1} \top \boxminus\!\!\rightarrow \perp$.*

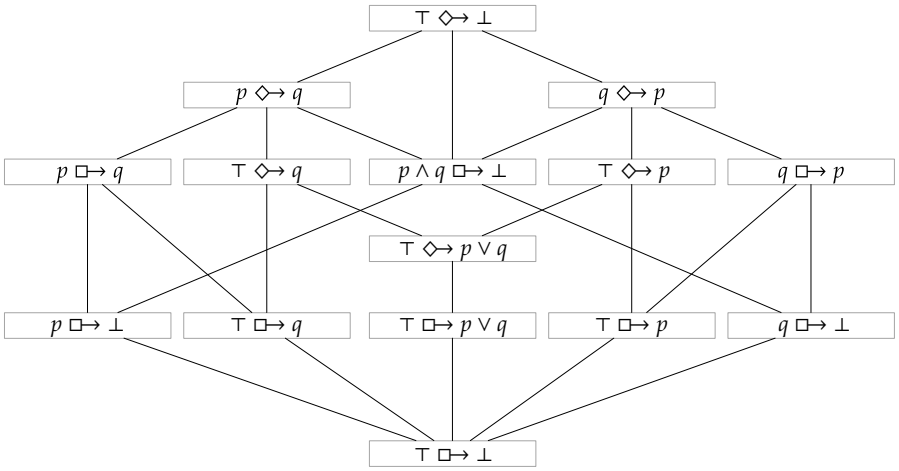

Fig. 1: The entailment lattice for $\{p, q\}$. If there is a line between two nodes, then the lower node entails the higher node.

3.5 Inference rules

The inference rules for KL_1 are given in Figure 2.

Note that, since the left- and right-hand sides of rules are sets of atoms, not sequences or multisets, a finite set R of rules has only a *finite* number of inferential consequences.

Given a set R of rules, the derived rules $deriv(R)$ are the rules generated by repeated application of the inference rules in Figure 2. We also write $R \vdash_{\mathrm{KL}_1} r$ if $r \in deriv(R)$.

Example 2 One can check that $\{p \square\!\!\rightarrow q,\ p \diamondsuit\!\!\rightarrow r\} \models_{\mathrm{KL}_1} p \diamondsuit\!\!\rightarrow p \vee q \vee r$. Here is a derivation using the inference rules:

$$\dfrac{\dfrac{\dfrac{p \square\!\!\rightarrow q}{p \diamondsuit\!\!\rightarrow q}\,\text{MAY-MUST} \qquad p \diamondsuit\!\!\rightarrow r}{p \diamondsuit\!\!\rightarrow q \vee r}\,\text{MAY-UNION} \qquad \dfrac{-}{p \diamondsuit\!\!\rightarrow p}\,\text{MAY-ID}}{p \diamondsuit\!\!\rightarrow p \vee q \vee r}\,\text{MAY-UNION}$$

Example 3 It can also be confirmed that $\{p \square\!\!\rightarrow q \vee r,\ q \square\!\!\rightarrow \bot\} \models_{\mathrm{KL}_1} p \square\!\!\rightarrow r$. Here is a derivation using the inference rules:

$$\dfrac{\dfrac{\dfrac{q \square\!\!\rightarrow \bot}{q \square\!\!\rightarrow r}\,\text{QUOD-LIBET}}{p \wedge q \square\!\!\rightarrow r}\,\text{MUST-SI} \qquad \dfrac{\dfrac{-}{r \square\!\!\rightarrow r}\,\text{MUST-ID}}{p \wedge r \square\!\!\rightarrow r}\,\text{MUST-SI} \qquad p \square\!\!\rightarrow q \vee r}{p \square\!\!\rightarrow r}\,\text{MUST-TRANS}$$

Proposition 5 (Soundness) *KL_1 is sound: $R \vdash_{KL_1} r$ implies $R \models_{KL_1} r$. That is, $deriv_1(R) \subseteq kl_1(R)$.*

$$\text{MUST-ID}\ \frac{-}{A \Box\!\!\rightarrow A}\ A \neq \emptyset \qquad \text{MUST-SI}\ \frac{A \Box\!\!\rightarrow B}{A' \Box\!\!\rightarrow B}\ A \subset A'$$

$$\text{MUST-UNION}\ \frac{A \Box\!\!\rightarrow B \quad A \Diamond\!\!\rightarrow C}{A \Box\!\!\rightarrow B \cup C}$$

$$\text{MUST-TRANS}\ \frac{A \Box\!\!\rightarrow b_1 \vee \ldots \vee b_n \quad A \wedge b_1 \Box\!\!\rightarrow C \quad \ldots \quad A \wedge b_n \Box\!\!\rightarrow C}{A \Box\!\!\rightarrow C}$$

$$\text{QUOD-LIBET}\ \frac{A \Box\!\!\rightarrow \bot}{A \Box\!\!\rightarrow B}$$

$$\text{MAY-ID}\ \frac{-}{A \Diamond\!\!\rightarrow A} \qquad \text{MAY-SI}\ \frac{A \Diamond\!\!\rightarrow B}{A' \Diamond\!\!\rightarrow B}\ A \subset A'$$

$$\text{MAY-UNION}\ \frac{A \Diamond\!\!\rightarrow B \quad A \Diamond\!\!\rightarrow C}{A \Diamond\!\!\rightarrow B \cup C}$$

$$\text{MAY-TRANS}\ \frac{A \Diamond\!\!\rightarrow b_1 \vee \ldots \vee b_n \quad A \wedge b_1 \Diamond\!\!\rightarrow C \quad \ldots \quad A \wedge b_n \Diamond\!\!\rightarrow C}{A \Diamond\!\!\rightarrow C}$$

if for every $c \in C$, $A \wedge c \Box\!\!\rightarrow b_i$ for some $b_i \in \{b_1, \ldots, b_n\}$

$$\text{MAY-SO}\ \frac{A \Diamond\!\!\rightarrow B}{A \Diamond\!\!\rightarrow B'}\ B' \subset B$$

$$\text{MAY-MUST}\ \frac{A \Box\!\!\rightarrow B}{A \Diamond\!\!\rightarrow B} \qquad \text{MAY-FALSUM}\ \frac{A \wedge b \Box\!\!\rightarrow \bot}{A \Diamond\!\!\rightarrow b}$$

Fig. 2: Inference rules of KL_1

Proof We show that if $r \in deriv_1(R)$ then, for all A, $out_1(R, A) = out_1(R \cup \{r\}, A)$. The proof is by induction on the length of the derivation. The base case is trivial. For the inductive step, suppose r was derived in one step from r_1 and r_2. By the inductive hypothesis $out_1(R \cup \{r_1, r_2\}, A) = out_1(R, A)$ for all A. We must show that, for all A:

$$out_1(R \cup \{r_1, r_2, r\},\ A) = out_1(R \cup \{r_1, r_2\},\ A)$$

By Proposition 1, if $X \in out_1(R \cup \{r_1, r_2, r\},\ A)$ then $X = \mathrm{M}(D, A)$ for some definite program $D \in def(R \cup \{r_1, r_2, r\})$ and $X \models R \cup \{r_1, r_2, r\}$.

Suppose r was derived from r_1 and r_2 using MUST-UNION. (The other cases are similar and are omitted[22]). Let $r_1 = A \Box\!\!\rightarrow B$, $r_2 = A \Diamond\!\!\rightarrow C$, $r = A \Box\!\!\rightarrow B \cup C$.

[22] For other inference rules it is not always the case that $def(\{r_1, r_2, r\}) = def(\{r_1, r_2\})$. However, for the first half of the proof it is sufficient to show that for every $D \in def(\{r1, r, 2, r\})$ there exists

It can be checked that $def(\{A \boxdot\!\!\rightarrow B \cup C\}) = def(\{A \boxdot\!\!\rightarrow B,\ A \diamond\!\!\rightarrow C\})$, and so:

$$def(R \cup \{r_1, r_2, r\}) = def(R \cup \{r_1, r_2\})$$

This establishes that $out_1(R \cup \{r_1, r_2, r\},\ A) \subseteq out_1(R \cup \{r_1, r_2\},\ A)$ for all A. To show the other inclusion, that $out_1(R \cup \{r_1, r_2\}\ A) \subseteq out_1(R \cup \{r_1, r_2, r\},\ A)$, it remains to show that if $X \models R \cup \{r_1, r_2\}$ then $X \models R \cup \{r_1, r_2, r\}$. To see this, note that if $X \models A \boxdot\!\!\rightarrow B$ then $X \models A \boxdot\!\!\rightarrow B \cup C$. □

Conjecture 1 (Completeness) KL_1 is complete: $R \models_{KL_1} r$ implies $R \vdash_{KL_1} r$. That is, $kl_1(R) \subseteq deriv_1(R)$.

We do not have a proof of this although we have strong reasons to believe it is true. Moreover we have tested it empirically using a computer implementation[23] of the definitions and results in Sections 3, 4, and 5. For KL_1, we sample randomly generated sets R of KL_1 rules, and individual rules r such that $R \models_{KL_1} r$. Then we generate all inferential consequences of R (always finite for a finite set of rules) and test if r is among the consequences. Extensive empirical testing, from sample sets of the order of 100,000 rules, suggests that KL_1 is indeed complete. We would of course prefer the full confidence of a formal proof.

3.5.1 A comparison between $\boxdot\!\!\rightarrow$ and $\diamond\!\!\rightarrow$ inference rules

The MUST-TRANS and MAY-TRANS rules are very similar. But there is one extra condition in MAY-TRANS which does not appear in MUST-TRANS. The reason for the extra condition is this. Consider the simpler variant:

$$\text{MAY-TRANS-BAD}\ \frac{A \diamond\!\!\rightarrow b_1 \vee \ldots \vee b_n \quad A \wedge b_1 \diamond\!\!\rightarrow C \quad \ldots \quad A \wedge b_n \diamond\!\!\rightarrow C}{A \diamond\!\!\rightarrow C}$$

This rule is exactly parallel to MUST-TRANS but it is unsound: it would allow us to infer $p \diamond\!\!\rightarrow r$ from $p \diamond\!\!\rightarrow q$ and $q \diamond\!\!\rightarrow r$.

Let $p \diamond\!\!\rightarrow q$ be "if you visit the south of France, then feel free to enter a naturist resort!". Let $q \diamond\!\!\rightarrow r$ be "if you enter a naturist resort, then feel free to disrobe entirely!". We do not want to infer "if you visit the south of France, then feel free to disrobe entirely!".

To avoid this, there is an extra condition in MAY-TRANS that insists that each c in C requires some b_i in $\{b_1, \ldots, b_n\}$: for every $c \in C$, there must be a rule $A \wedge c \boxdot\!\!\rightarrow b_i$ for some $b_i \in \{b_1, \ldots, b_n\}$.

$D' \in def(\{r1, r, 2\})$ such that D and D' have the same models (and therefore the same minimal model). This is straightforward for all the inference rules of Figure 2.

[23] See `https://github.com/RichardEvans/kl_haskell`.

3.6 The primary and secondary aim of logic

In Section 2.4 above, we claimed that, for Kant, transcendental logic's primary question is: given a set of subsumptions, and a set of rules, what further subsumptions may/must I perform? This question is answered by the out_1 function defined in Section 3.2 above and the further characterised by the results of the sections that follow.

Transcendental logic's secondary question is: given a set of rules I have adopted, what further rules may/must I adopt? This question is answered by the rule entailment ($\models_{\mathrm{KL}_1}$) and inference rules ($\vdash_{\mathrm{KL}_1}$) defined in Sections 3.4 and 3.5 above.

3.7 Comparing KL_1 with other input / output logics

The family of input/output logics, broadly conceived, are logics that operate on elements that do not (necessarily) have truth-values, and in which inferences are not closed under contraposition. KL_1 is a member of the family of input/output logics, very broadly conceived. However, there are a number of essential differences between KL_1 and the particular input/output logics described in [41].

There is the obvious notational difference. Standard input/output logics use the pair notation (p, q) to indicate implication from p to q. That in itself is a trivial difference but KL_1 needs two distinct arrows, $p \boxdot\!\!\rightarrow q$ and $p \Diamond\!\!\rightarrow q$, for the two different types of rule (see Section 2.2).

That aside, the first major difference is that, in the standard input/output logics, a rule (ϕ, ψ) relates arbitrarily complex expressions that are closed under the Boolean connectives. In KL_1, by contrast, the elements are conjunctions and disjunctions of *atoms* only.

In KL_1, the left hand side (antecedent) of a rule is a set of atoms representing a conjunction, while the right hand side (consequent) is a sets of atoms representing a disjunction. In a standard input/output logic, if we apply the rule $(p, q \vee r)$ to the premises $\{p\}$, we get the single result $\{q \vee r\}$. In KL_1, by contrast, if we apply the rule $p \boxdot\!\!\rightarrow q \vee r$ to the premises $\{p\}$, we get three possible answers:

$$out_1(\{p \boxdot\!\!\rightarrow q \vee r\}, \{p\}) = \{\{p, q\}, \{p, r\}, \{p, q, r\}\}$$

The second major difference, then, is that the output out_1 of a set of rules in KL_1 is a set of sets of atoms, representing different possible ways to satisfy the rules, while the output in a standard input/output logic is a single set of propositions.

A third major difference is that KL_1 does not have an inference rule for weakening the output. Although there is a rule for strengthening the input (MUST-SI), there is no corresponding rule for weakening the output (WO):

$$\text{WO}\ \frac{A \boxdot\!\!\rightarrow B}{A \boxdot\!\!\rightarrow B \cup C}$$

This rule is not valid in KL_1 because it would license activities that were arbitrary[24]. Suppose, for example, the agent is performing the action p and has the rule:

$$p \Box\!\!\rightarrow q$$

If we are allowed to infer, using WO, that

$$p \Box\!\!\rightarrow q \vee r$$

then there will be two possible sets of actions that are compatible with the original rule plus the derived rule: $\{p, q\}$ and $\{p, q, r\}$. The trouble is that the action r that is introduced in the second answer is *arbitrary* in the sense that it is not itself grounded in a rule.

Throughout his mature writings, Kant assumes that all activity must be grounded in a rule in order to count as activity at all. Consider the following difference: some of the movements my body performs are mere spasms, while other movements count as actions. The difference between the two, according to Kant, is that actions are movements that are grounded in a rule I have adopted. Kant's fundamental normative step is to characterise the subset of my bodily movements that count as actions as those which are subsumed under a rule. This is as true of mental activity as it is of physical activity:

> Like all our powers, the understanding in particular is bound in its actions to rules [*Jäsche Logic* §1] [25]

Therefore, in any logic that tries to capture Kant's normative theory of mental activity, weakening output (WO) should not be valid (see Section 3.9). KL_1 has only the following restricted form:

$$\text{MUST-UNION}\ \frac{A \Box\!\!\rightarrow B \quad A \Diamond\!\!\rightarrow C}{A \Box\!\!\rightarrow B \cup C}$$

The final difference between KL_1 and the standard input/output logics is that KL_1 allows what is called "throughput" in input/output logics: for each $X \in out_1(R, A)$, $A \subseteq X$. The inference rule corresponding to throughput is MUST-ID, allowing us to infer $p \Box\!\!\rightarrow p$: "if you are doing p, then do p!", for all actions p. The reason why this inference rule is valid in KL_1 is again because of the particular intended application: KL_1 is a logic of concurrent activity, prescribing the activities that we must perform conditional on the activities we are already performing. It is for the agent to produce a package of activities that together satisfy the various rules. If it is already performing action p, then p must be part of any complete package of activity. If you are already doing p, there is no point trying to undo the performance of p – that ship has already sailed.

[24] [1] and [2] make a similar point, for different reasons. They argue that $B \vee C$ may not constitute a "whole". See footnotes 10 and 41 of [1].

[25] Or to put it another way, mental occurrences not grounded in a rule "would then belong to no experience, and would consequently be without an object, and would be nothing but a blind play of representations, i.e., less than a dream." [A112], see also [A156/B195]. See [18] for further elaboration.

3.8 Comparing KL_1 to other logics of imperatives

Many logics of imperatives[26] start with a base truth-functional logic and extend it with one or more imperatival operators (e.g. "!"). For example, given a set Σ of sentences that have a truth-functional semantics (e.g. sentences of classical propositional logic or first-order logic), with S ranging over sentences in Σ, define an imperative language L as:

$$L := S \mid !S$$

Given such a framework, imperatival inference can be explained using Dubislav's trick [25]: $!p$ entails $!q$ whenever p entails q.

We stress, however, that KL_1 does not use this framework. We do not presuppose an existing base language Σ in which truth-conditions have already been assigned[27]. In KL_1, the atoms $\mathcal{A}$ represent actions that do not have truth-values. In this crucial respect, KL_1 is closer to the input/output logics than it is to most logics of imperatives.

[12] identifies three sets of requirements that any logic of imperatives must satisfy:

1. imperatives can stand in inconsistency relations
2. imperatives can stand in inferential relations
3. imperatives can be embedded

Requirement 1: A set R of rules is strongly inconsistent in KL_1 if, for every set A of atoms, $out_1(R, A) = \emptyset$. A set R of rules is **weakly inconsistent** if there is *some* set A of atoms such that $out_1(R, A) = \emptyset$.

Example 4

$$R_1 = \begin{cases} \top \boxdot\!\!\rightarrow p \\ p \boxdot\!\!\rightarrow \bot \end{cases} \qquad R_2 = \begin{cases} p \boxdot\!\!\rightarrow q \\ p \boxdot\!\!\rightarrow r \\ q \wedge r \boxdot\!\!\rightarrow \bot \end{cases}$$

Here, R_1 is strongly inconsistent but R_2 is only weakly inconsistent. When $A = \{\}$, $out_1(R_2, \{\}) = \{\emptyset\} \neq \emptyset$.

[12] insists that an imperative requiring ϕ is inconsistent with a permissive allowing $\sim\phi$. In KL_2, where we add a form of negation to KL_1, the following set R_5 is not inconsistent (not even weakly inconsistent):

$$R_5 = \begin{cases} p \boxdot\!\!\rightarrow q \\ p \diamond\!\!\rightarrow \sim q \end{cases}$$

This is because the permissive $\diamond\!\!\rightarrow$ rule is *weak* and is overridden by a $\boxdot\!\!\rightarrow$ rule.

[26] For example, [13,27,12].

[27] In this respect, our approach follows the pragmatist order of explanation, in which rules specifying what to do are explanatorily prior to rules specifying what is the case. See Sections 2.4 and 6.

Of course, differences of intuition are to be expected here because [12] is developing a semantics for conditional imperatives in natural language, while our project is to provide a logic of conditional imperatives for describing *rules of thought*.

Requirement 2: inferential relations in KL_1 are defined in terms of a semantic ($\models_{KL_1}$) and syntactic ($\vdash_{KL_1}$) notion of entailment. We have commented on the relationship (soundness and completeness) between them.

Requirement 3: the central motivating cases of embedding in [12] are conditional imperatives and permissives. These are precisely the types of imperative that KL_1 is designed to model. Although we agree that other forms of embeddedness are also important, we do not have space to do justice to a full discussion here. We have developed an extension of KL_1 that includes embedded rules (e.g. $(p \boxdot\!\!\rightarrow q) \boxdot\!\!\rightarrow r$), but we leave a full description to further work.

3.9 The deontic paradoxes in KL_1

Ross's paradox [46] was first described for von Wright's deontic logic, but it also applies to logics of conditional imperatives. Suppose we are given the order:

Post the letter!

Now the declarative proposition "x posts the letter" entails the proposition "either x posts the letter or x burns it." But we do not want to infer from this entailment and the original order that:

Therefore: post the letter or burn it!

One way of seeing the problem with this conclusion is by inferring (via the inference that if you must do something, then you may do it):

Therefore: you may post the letter or burn it!

and then inferring (since permission distributes over disjunctions):

Therefore: you may burn it!

The absurdity in this chain of reasoning comes out even more clearly if the newly introduced disjunct is something altogether irrelevant to posting the letter, and altogether unacceptable. For example:

Post the letter!
Therefore: post the letter or set fire to the school!
Therefore: you may post the letter or set fire to the school!
Therefore: you may set fire to the school!

In the usual input/output logics, the rule for weakening the output is:

$$\text{WO}\ \frac{A \boxdot\!\!\rightarrow B}{A \boxdot\!\!\rightarrow B \vee C}$$

KL_1 avoids this paradox because it does not have the troublesome WO inference rule (see Section 3.7).

There is a paradox that is closely related to Ross's paradox, that involves conjunction rather than disjunction. Suppose you are permitted to wipe your feet and enter the house. It does not follow that you are permitted to enter the house *simpliciter*[28]. This troublesome inference does not go through in KL_1. Letting w stand for "wipe your feet", e stand for "enter the house", and c stand for the conjunction of both activities, the conjunctive permission is represented by the set R of rules[29]:

$$R = \begin{cases} \top \diamondsuit\!\!\rightarrow c \\ c \boxdot\!\!\rightarrow w \\ c \boxdot\!\!\rightarrow e \end{cases}$$

Here there are two acceptable packages of actions: doing nothing, or doing both w and e:

$$out_1(R, \{\}) = \begin{cases} \{\} \\ \{e, w, c\} \end{cases}$$

Note that neither $\{e\}$ nor $\{c, e\}$ are elements of $out_1(R, \{\})$.

There is a third, related paradox involving implication. Suppose we have:

> You must do a or b!
> If you do a then you must do c!

We do not want to infer from these rules that:

> You may do b and c!

The trouble with this inference is that doing c is only conditionally permissible: it is only if you are doing a that you must (and hence may) do c. Consider the concrete example:

> You must either leave the dinner early or stay until the end.
> If you leave early, then you must interrupt the conversation to tell everyone you are going early.

[28] See the related "Window paradox" in [25].

[29] It is straightforward to extend the form of rules in KL_1 to allow disjunctions of conjunctions of atoms in the heads (consequents) of rules. In that version, the example would be represented by $R' = \{\top \boxdot\!\!\rightarrow (w \wedge e)\}$ with $out_1(R', \{\}) = \{\{\}, \{w.e\}\}$. The details are straightforward but we have omitted them here to shorten the presentation.

It would not be acceptable to both stay until the end but also interrupt the conversation to tell everyone that you were leaving. This troublesome inference does *not* go through in KL_1:

$$out_1(\{\top \boxdot\!\!\rightarrow a \vee b,\ a \boxdot\!\!\rightarrow c\}, \{\}) = \begin{cases} \{a, c\} \\ \{b\} \\ \{a, b, c\} \end{cases}$$

Note that $\{b, c\}$ is not an element of $out_1(\{\top \boxdot\!\!\rightarrow a \vee b,\ a \boxdot\!\!\rightarrow c\}, \{\})$.

4 KL_2: extending KL_1 with negation

We define KL_2 by adding a negation operator to KL_1. This negation operator applies to an atom a to generate a literal $\sim a$. It can only be applied to atoms; we do not allow expressions such as $\sim\sim p$, $\sim(p \wedge q)$ or $\sim(p \vee q)$.

Henceforth, a, b, c range over literals and A, B, C and X range over sets of literals. Where c is a literal we write $\overline{c}$ for the complement of c: if c is an atom then $\overline{c} = \sim c$ and $\overline{\sim c} = c$. When C is a set of literals $\overline{C} = \{\overline{c} \mid c \in C\}$.

The rules of KL_2 are:

$$\mathcal{R} ::= B \Box\!\!\rightarrow C \mid B \Diamond\!\!\rightarrow C$$

as for KL_1 except that now B and C are sets of literals (representing conjunctions and disjunctions respectively).

A set of literals X satisfies a set R of rules, $X \models R$, and satisfies an individual rule r, $X \models r$, just as in KL_1 except that now the elements of a rule are sets of literals rather than atoms: $X \models B \Diamond\!\!\rightarrow C$ always; $X \models B \Box\!\!\rightarrow C$ if $B \not\subseteq X$ or $C \cap X \neq \emptyset$.

4.1 Minimal requirements on a Kantian negation operator

Kant describes a variety of properties that negation must satisfy throughout the *Jäsche Logic*[30]. As minimal requirements, we pick out two fundamental properties that he insists on.

The operator $\sim$ from atoms to literals is our negation operator. First, p and $\sim p$ must be incompatible[31]. Second, $\sim p$ must be the *most general* proposition that is incompatible with p[32]: for any q, if p and q are incompatible, then q must entail $\sim p$.

To motivate the second requirement, consider the following example. Suppose Jill can support at most one of three football teams: Arsenal, Barnet, or Chelsea. She cannot support more than one: supporting Barnet is incompatible with supporting Arsenal. But "Jill supports Barnet" (b) cannot be the negation of "Jill supports Arsenal" (a) because it is *too specific*. The negation $\sim a$ of "Jill supports Arsenal" is the most general claim that is incompatible with her supporting Arsenal, and "Jill supports Chelsea" (c) is also incompatible with her supporting Arsenal. All we can say about $\sim a$ is that b entails $\sim a$ and c entails $\sim a$.

[30] See [*Jäsche Logic* p.51, 103-4, 109, 117-19, 124ff.].

[31] See the principle of contradiction in [A150ff./B189ff.] and [*Jäsche Logic* p.51].

[32] Kant makes this precise claim in [*Jäsche Logic* §49]: "one of [a pair of contrary judgements] says more . . . than the mere negation of the other." In other words, a claim that is incompatible with p entails (but is not necessarily entailed by) the negation of p. See also Brandom [7], Humberstone [28], p.1170.

4.2 Inference rules

The extra inference rules for KL_2 are given in Figure 3. They are chosen to capture the two requirements on the negation operator described above. Here we are assuming that the mutual incompatibility of a (non-empty) set A of literals is expressed by the rule $A \boxdot\!\!\rightarrow \bot$.

$$\sim\text{-LEFT}\ \frac{-}{c \wedge \sim c \boxdot\!\!\rightarrow \bot} \qquad \sim\text{-RIGHT}\ \frac{A \wedge b \boxdot\!\!\rightarrow \bot}{A \boxdot\!\!\rightarrow \overline{b}}$$

Fig. 3: Additional inference rules for KL_2

Example 5 Here we use $p \boxdot\!\!\rightarrow q$ to derive $\sim q \boxdot\!\!\rightarrow \sim p$:

$$\dfrac{\dfrac{\dfrac{p \boxdot\!\!\rightarrow q}{p \wedge \sim q \boxdot\!\!\rightarrow q}\ \text{MUST-SI} \qquad \dfrac{\dfrac{}{q \wedge \sim q \boxdot\!\!\rightarrow \bot}\ \sim\text{-LEFT}}{p \wedge q \wedge \sim q \boxdot\!\!\rightarrow \bot}\ \text{MUST-SI}}{p \wedge \sim q \boxdot\!\!\rightarrow \bot}\ \text{MUST-TRANS}}{\sim q \boxdot\!\!\rightarrow \sim p}\ \sim\text{-RIGHT}$$

Example 6 Here we derive $\top \boxdot\!\!\rightarrow \sim q$ from $p \wedge q \boxdot\!\!\rightarrow \bot$ and $\sim p \wedge q \boxdot\!\!\rightarrow \bot$:

$$\dfrac{\dfrac{\dfrac{p \wedge q \boxdot\!\!\rightarrow \bot}{q \boxdot\!\!\rightarrow \sim p}\ \sim\text{-RIGHT} \qquad \sim p \wedge q \boxdot\!\!\rightarrow \bot}{q \boxdot\!\!\rightarrow \bot}\ \text{MUST-TRANS}}{\top \boxdot\!\!\rightarrow \sim q}\ \sim\text{-RIGHT}$$

It is instructive to look at some derived rules of KL_2. Those in Figure 4 are all derivable using only $\sim$-LEFT and the rules of KL_1. TRANSPOSE is obtained from $\sim$-LEFT using MUST-SI and MUST-TRANS. MUST-$\bot$ is obtained by repeated application of TRANSPOSE. (The case of MUST-$\bot$ where $C = \emptyset$ is vacuous but harmless.) INCONS generalises $\sim$-LEFT. We do not show the derivations in detail. They are very straightforward and will be shown in detail when we present their KL_3 versions in Section 5.

Of particular interest is the following pair:

$$\text{REDUCE-}\bot\ \frac{B \wedge c \boxdot\!\!\rightarrow \bot \quad B \wedge \sim c \boxdot\!\!\rightarrow \bot}{B \boxdot\!\!\rightarrow \bot} \qquad \text{RESOLVE-}\bot\ \frac{A \wedge c \boxdot\!\!\rightarrow \bot \quad B \wedge \sim c \boxdot\!\!\rightarrow \bot}{A \cup B \boxdot\!\!\rightarrow \bot}$$

The first is a special case of the second. They will be discussed in more detail in the treatment of KL_3. Unlike the inference rules in Figure 4 their derivation relies on $\sim$-RIGHT.

Example 7 Here is how the earlier *Examples 5* and *6* look with these derived inference rules.

To derive $\sim q \boxdot\!\!\rightarrow \sim p$ from $p \boxdot\!\!\rightarrow q$:

$$\text{MUST-}\bot\ \frac{B \boxdot\!\!\rightarrow C}{B \cup \overline{C} \boxdot\!\!\rightarrow \bot} \qquad \text{TRANSPOSE}\ \frac{B \boxdot\!\!\rightarrow b \vee C}{B \wedge \overline{b} \boxdot\!\!\rightarrow C}$$

$$\text{INCONS}\ \frac{A \boxdot\!\!\rightarrow c \quad B \boxdot\!\!\rightarrow {\sim}c}{A \cup B \boxdot\!\!\rightarrow \bot}$$

Fig. 4: Some derived inference rules for KL_2

$$\dfrac{\dfrac{p \boxdot\!\!\rightarrow q}{p \wedge {\sim}q \boxdot\!\!\rightarrow \bot}\ \text{MUST-}\bot}{{\sim}q \boxdot\!\!\rightarrow {\sim}p}\ {\sim}\text{-RIGHT}$$

To derive $\top \boxdot\!\!\rightarrow {\sim}q$ from $p \wedge q \boxdot\!\!\rightarrow \bot$ and ${\sim}p \wedge q \boxdot\!\!\rightarrow \bot$:

$$\dfrac{\dfrac{\dfrac{p \wedge q \boxdot\!\!\rightarrow \bot}{q \boxdot\!\!\rightarrow {\sim}p}\ {\sim}\text{-RIGHT} \qquad \dfrac{{\sim}p \wedge q \boxdot\!\!\rightarrow \bot}{q \boxdot\!\!\rightarrow p}\ {\sim}\text{-RIGHT}}{q \boxdot\!\!\rightarrow \bot}\ \text{INCONS}}{\top \boxdot\!\!\rightarrow {\sim}q}\ {\sim}\text{-RIGHT}$$

With RESOLVE-$\bot$ it is even easier:

$$\dfrac{\dfrac{p \wedge q \boxdot\!\!\rightarrow \bot \qquad {\sim}p \wedge q \boxdot\!\!\rightarrow \bot}{q \boxdot\!\!\rightarrow \bot}\ \text{RESOLVE-}\bot}{\top \boxdot\!\!\rightarrow {\sim}q}\ {\sim}\text{-RIGHT}$$

4.3 Semantics

A set X of literals is **consistent** if it does not contain a pair of complementary literals a and ${\sim}a$ for any atom a. It is inconsistent otherwise.

$\mathcal{A}$ denotes the set of atoms. Let $C_{\mathcal{A}}$ denote the set of constraint rules

$$C_{\mathcal{A}} = \{a \wedge {\sim}a \boxdot\!\!\rightarrow \bot \mid a \in \mathcal{A}\}$$

We omit the subscript $\mathcal{A}$ where it is obvious from context. Clearly the set X of literals is consistent when $X \models C$.

We write $V_{\mathcal{A}}$ for the set of maximal consistent sets of literals from $\mathcal{A}$, i.e., $V_{\mathcal{A}}$ is the set of sets X_m such that X_m is consistent and, for every $a \in \mathcal{A}$, either $a \in X_m$ or ${\sim}a \in X_m$.

Definition 6 Given a set R of rules, a set X of literals is a **violating set** of R if there is no maximal consistent $X_m \in V_{\mathcal{A}}$ such that $X_m \supseteq X$ and $X_m \models R$.

One can see that an inconsistent set of literals is, by definition, a violating set of every set of rules. And if X is a violating set of R then so is every $X' \supseteq X$.

A violating set X of R cannot be extended to a maximal consistent set $X_m \supseteq X$ that satisfies every rule in R. If $B \boxdot\!\!\rightarrow \bot$ is a rule in R then X is a violating set of R when $B \subseteq X$. For a rule of the form $B \boxdot\!\!\rightarrow C$ ($C \neq \emptyset$) and without the consistency requirement, a set X of literals can always be extended to a set $X' \supseteq X$ that satisfies that rule (the set of all literals satisfies it). With the consistency requirement, X cannot be extended to a consistent $X' \supseteq X$ that satisfies $B \boxdot\!\!\rightarrow C$ when $B \cup \overline{C} \subseteq X$. (Indeed that condition applies to constraint rules also: X cannot be extended to satisfy $B \boxdot\!\!\rightarrow \bot$ when $B \cup \overline{\emptyset} \subseteq X$.) It is possible that X is a violating set of a set R of rules even though X is not a violating set of any of them individually. A rule $B \diamond\!\!\rightarrow C$ is satisfied by every set X of literals: only inconsistent sets of literals are violating sets of $\diamond\!\!\rightarrow$ rules.

Example 8

$$R = \{\, p \boxdot\!\!\rightarrow q,\ p \boxdot\!\!\rightarrow {\sim}q \,\}$$

The consistent sets $\{p\}$, $\{p, {\sim}q\}$ and $\{p, q\}$ are violating sets of R. All inconsistent sets are also violating sets of R.

Compare:

$$R' = \{\, p \wedge q \boxdot\!\!\rightarrow \bot,\ p \wedge {\sim}q \boxdot\!\!\rightarrow \bot \,\}$$

Again, $\{p\}$, $\{p, {\sim}q\}$ and $\{p, q\}$ are violating sets of R'.

Example 9

$$R'' = \{\, p \wedge q \boxdot\!\!\rightarrow r,\ p \wedge {\sim}q \boxdot\!\!\rightarrow r,\ p \wedge r \boxdot\!\!\rightarrow \bot \,\}$$

$\{p\}$ is a violating set of R''.

Example 10

$$R''' = \{\, p \wedge q \boxdot\!\!\rightarrow r,\ p \wedge {\sim}q \boxdot\!\!\rightarrow r,\ r \wedge s \boxdot\!\!\rightarrow \bot \,\}$$

Here $\{r, s\}$, $\{p, s\}$ and $\{p, {\sim}r\}$ are the minimal consistent violating sets.

Definition 7 Given a set R of rules, we define an auxiliary set of rules $aux(R)$. The outcome function out_2 for KL_2 is defined using $aux(R)$.

$$aux(R) \;=\; \{A - \{a\} \boxdot\!\!\rightarrow \overline{a} \mid A \text{ is a finite violating set of } R,\ a \in A\}$$
$$out_2(R, A) \;=\; out_1(R \cup C \cup aux(R), A)$$

Example 11 If $R = \{p \boxdot\!\!\rightarrow \bot\}$, then $\{p\}$ and $\{p, {\sim}p\}$ are violating sets of R, and

$$aux(R) = \{\, \top \boxdot\!\!\rightarrow {\sim}p,\ p \boxdot\!\!\rightarrow p,\ {\sim}p \boxdot\!\!\rightarrow {\sim}p,\ \ldots \}$$

Note the close parallel between the inference rules and the semantics. In $out_2(R, A)$, the check for consistency, expressed by the rules C, matches ∼-LEFT, while the set of rules $aux(R)$ provides the additional inferences afforded by ∼-RIGHT. Indeed, we will show (below) that X is a (finite) violating set of R if, and only if, the rule $X \boxdot\!\!\rightarrow \bot$ is entailed by R in KL_2. That will establish the connections between semantic and syntactic entailment in KL_2.

Rule entailment is defined as in KL_1.

Definition 8 Two rule sets R_1 and R_2 are **strongly equivalent** in KL_2 if:

$$out_2(R_1, A) = out_2(R_2, A) \quad \text{for all sets } A \text{ of atoms}$$

A set R of rules **entails** a rule r in KL_2, written $R \models_{\mathrm{KL}_2} r$, if R and $R \cup \{r\}$ are strongly equivalent in KL_2.

We write $kl_2(R)$ for the set of rules semantically entailed by R in KL_2: $kl_2(R) = \{r \mid R \models_{\mathrm{KL}_2} r\}$. Rule sets R_1 and R_2 are strongly equivalent in KL_2 when $kl_2(R_1) = kl_2(R_2)$.

For brevity, we will write $\mathrm{KL}_1(C)$ for KL_1 extended with the inference rule $\sim$-LEFT and say that R $\mathrm{KL}_1(C)$-entails r when $R \cup C \models_{\mathrm{KL}_1} r$.

Proposition 6 (Decomposition) *A set R of rules semantically entails a rule r in KL_2 if and only if $R \cup C \cup aux(R)$ semantically entails r in KL_1. That is, for all rule sets R:*

$$kl_2(R) = kl_1(R \cup C \cup aux(R))$$

Proof $kl_2(R) = kl_1(R \cup C \cup aux(R))$ for all R is equivalent to saying that two rules sets R_1 and R_2 are strongly equivalent in KL_2, $kl_2(R_1) = kl_2(R_2)$, iff $R_1 \cup C \cup aux(R_1)$ and $R_2 \cup C \cup aux(R_2)$ are strongly equivalent in KL_1, $kl_1(R_1 \cup C \cup aux(R_1)) = kl_1(R_2 \cup C \cup aux(R_2))$.

$$\begin{aligned} kl_2(R_1) = kl_2(R_2) & \\ \text{iff} \quad & out_2(R_1, A) = out_2(R_2, A) \quad \text{for all } A \\ \text{iff} \quad & out_1(R_1 \cup C \cup aux(R_1),\ A) = out_1(R_2 \cup C \cup aux(R_2),\ A) \quad \text{for all } A \\ \text{iff} \quad & kl_1(R_1 \cup C \cup aux(R_1)) = kl_1(R_2 \cup C \cup aux(R_2)) \end{aligned}$$

□

The following is a general property of KL_1.

Proposition 7 *Let R be a set of rules and A a (finite) set of literals:*

$$out_1(R, A) = \emptyset \quad \textit{iff} \quad R \models_{KL_1} A \boxdot\!\!\rightarrow \bot$$

Proof We need to show that $out_1(R, A) = \emptyset$ iff $out_1(R \cup \{A \boxdot\!\!\rightarrow \bot\},\ A') = out_1(R, A')$ for all sets A' of literals.

For left-to-right: suppose $out_1(R, A) = \emptyset$. First observe that $out_1(R \cup \{A \boxdot\!\!\rightarrow \bot\},\ A') \subseteq out_1(R, A')$ for all A'. (Because if $X \in out_1(R \cup \{A \boxdot\!\!\rightarrow \bot\}, A')$ then X is computed from assumptions A' using the non-constraint rules of R and $X \models R \cup \{A \boxdot\!\!\rightarrow \bot\}$. Since $X \models R$, that means $X \in out_1(R, A')$ also.)

It remains to show that $out_1(R, A') \subseteq out_1(R \cup \{A \boxdot\!\!\rightarrow \bot\}, A')$ for all A'. Assume $X \in out_1(R, A')$; we shall show $X \in out_1(R \cup \{A \boxdot\!\!\rightarrow \bot\}, A')$. Since $X \in out_1(R, A')$, there is a D in $def(R)$ such that $X = \mathrm{M}(D, A')$. We will prove, first, that D is one of the definite programs of $R \cup \{A \boxdot\!\!\rightarrow \bot\}$, and second, that $X \models R \cup \{A \boxdot\!\!\rightarrow \bot\}$. First, since $def_r(A \boxdot\!\!\rightarrow \bot) = \{\emptyset\}$, $D \in def(R \cup \{A \boxdot\!\!\rightarrow \bot\})$. So D, as well as being one of the definite programs of R, is also one of the definite programs of $R \cup \{A \boxdot\!\!\rightarrow \bot\}$. Second, since $X \in out_1(R, A')$, $X \models R$. We just need

to show $X \models A \boxdot\!\!\rightarrow \bot$. Since $out_1(R, A) = \emptyset$, $A \not\models R$ by Proposition 2. Now, since $X \models R$, $A \not\subseteq X$, hence $X \models A \boxdot\!\!\rightarrow \bot$. These two claims entail, using Proposition 1, that $X \in out_1(R \cup \{A \boxdot\!\!\rightarrow \bot\}, A')$.

For the other direction: suppose $out_1(R, A) \neq \emptyset$. We need to show that there is some A' such that $out_1(R \cup \{A \boxdot\!\!\rightarrow \bot\},\ A') \neq out_1(R, A')$. Take $A' = A$: clearly $out_1(R \cup \{A \boxdot\!\!\rightarrow \bot\},\ A) = \emptyset$. □

It is a corollary of the above that R is strongly inconsistent in KL_1 if and only if $R \models_{\mathrm{KL}_1} \top \boxdot\!\!\rightarrow \bot$. This was Proposition 4.

Now we are ready to prove what we want, that X is a (finite) violating set of R precisely when $X \boxdot\!\!\rightarrow \bot$ is KL_2-entailed by R. Informally, if X is a (finite, non-empty) violating set of R then

$$\{X - \{a\} \boxdot\!\!\rightarrow \overline{a} \mid a \in X\} \subseteq aux(R)$$

And $aux(R) \subseteq kl_1(R \cup C \cup aux(R))$. So straight away:

$$\{X - \{a\} \boxdot\!\!\rightarrow \overline{a} \mid a \in X\} \subseteq kl_2(R)$$

Now $X \boxdot\!\!\rightarrow \bot \in kl_2(R)$ because (for any non-empty finite set X of literals) $X \boxdot\!\!\rightarrow \bot$ is entailed in $\mathrm{KL}_1(C)$ by $X - \{a\} \boxdot\!\!\rightarrow \overline{a}$, any $a \in X$. Syntactically, that is easy to see. It is just an instance of MUST-$\bot$:

$$\frac{B \boxdot\!\!\rightarrow c}{B \wedge \overline{c} \boxdot\!\!\rightarrow \bot}$$

which is a derived rule of $\mathrm{KL}_1(C)$. Semantically, we want to confirm that $B \wedge \overline{c} \boxdot\!\!\rightarrow \bot \in kl_1(\{B \boxdot\!\!\rightarrow c\} \cup C)$. That is very easy (see proof below).

Proposition 8 *Let R be a set of rules. If X is a finite violating set of R then:*

$$X \boxdot\!\!\rightarrow \bot \in kl_1(aux(R) \cup C)$$

Proof If $X = \emptyset$ ($\emptyset$ is a violating set of R) then $\{a\}$ and $\{\sim a\}$ are also violating sets of R, any atom a, and $\{\top \boxdot\!\!\rightarrow a,\ \top \boxdot\!\!\rightarrow \sim a\} \subseteq aux(R)$. Clearly $aux(R) \cup C$ is strongly inconsistent in KL_1 and so $aux(R) \cup C \models_{\mathrm{KL}_1} \top \boxdot\!\!\rightarrow \bot$ (Proposition 4).

Suppose $X \neq \emptyset$. If X is a (finite) violating set of R then $\{X - \{a\} \boxdot\!\!\rightarrow \overline{a} \mid a \in X\} \subseteq aux(R)$. We show that $X \boxdot\!\!\rightarrow \bot \in kl_1(C \cup aux(R))$ by showing that $out_1(C \cup \{X - \{a\} \boxdot\!\!\rightarrow \overline{a}\}),\ X) = \emptyset$.

Consider any rule $X - \{a\} \boxdot\!\!\rightarrow \overline{a}$, $a \in X$. Suppose, for contradiction, that $X' \in out_1(C \cup \{X - \{a\} \boxdot\!\!\rightarrow \overline{a}\}),\ X)$. Then $X' \supseteq X$ and $X' \models C \cup \{X - \{a\} \boxdot\!\!\rightarrow \overline{a}\}$. In order to satisfy $X - \{a\} \boxdot\!\!\rightarrow \overline{a}$, X' must contain $\overline{a}$. But $a \in X$ so X' also contains a, and $X' \not\models C$. □

Remark The proof above shows that if $\emptyset$ is a violating set of R then $aux(R) \cup C$ is strongly inconsistent in KL_1. We can also show that if $R \cup C$ is strongly inconsistent in KL_1 then $\emptyset$ is a violating set of R. (Because then $out_1(R \cup C, X) = \emptyset$ for every set X of literals including every maximal consistent set X, and $\emptyset$ is thus a violating set of R.) The converse however is not true. Consider $R = \{p \boxrightarrow \bot,\ {\sim}p \boxrightarrow \bot\}$. $\emptyset$ is a violating set of R but R is not strongly inconsistent: $out_1(R, \emptyset) = \{\emptyset\} \neq \emptyset$.

Proposition 9 *Let R be a set of rules and X a finite set of literals.*

$$R \models_{KL_2} X \boxrightarrow \bot \text{ iff } X \text{ is a violating set of } R$$

Proof One half follows from the preceding result: if X is a (finite) violating set of R then $X \boxrightarrow \bot \in kl_1(C \cup aux(R))$; $kl_1(C \cup aux(R)) \subseteq kl_1(R \cup C \cup aux(R))$ and so $X \boxrightarrow \bot \in kl_2(R)$. It remains to prove that if $X \boxrightarrow \bot \in kl_2(R)$ then X is a violating set of R. We will prove that if $out_1(R \cup C \cup aux(R),\ X) = \emptyset$ then X is a violating set of R.

Consider any definite logic program D_R in the encoding $def(R)$ of R. Let $def(aux(R)) = \{D_{aux}\}$ (all rules in $aux(R)$ are $\boxrightarrow$ rules with singleton heads and so there is a single definite program encoding $aux(R)$). $\mathrm{M}(D_R \cup D_{aux},\ X) \models aux(R)$ so it must be that $\mathrm{M}(D_R \cup D_{aux},\ X) \not\models R \cup C$, i.e., either $\mathrm{M}(D_R \cup D_{aux},\ X)$ is inconsistent or $B \subseteq \mathrm{M}(D_R \cup D_{aux},\ X)$ for some rule $B \boxrightarrow \bot$ in R.

If X is inconsistent then X is a violating set of R. If X is consistent then consider any maximal consistent $X_m \supseteq X$. $\mathrm{M}(D_R \cup D_{aux},\ X_m) \supseteq \mathrm{M}(D_R \cup D_{aux},\ X)$, and since X_m is maximal, $X_m \supseteq \mathrm{M}(D_R \cup D_{aux},\ X_m) \supseteq \mathrm{M}(D_R \cup D_{aux},\ X)$. If $\mathrm{M}(D_R \cup D_{aux},\ X)$ is inconsistent then so is X_m, and that cannot be. So $B \subseteq \mathrm{M}(D_R \cup D_{aux},\ X)$ for some rule $B \boxrightarrow \bot$ in R. But then $B \subseteq X_m$, and $X_m \not\models R$. □

Note that according to the above, $\emptyset$ is a violating set of R if and only if $R \models_{\mathrm{KL}_2} \top \boxrightarrow \bot$.

Two refinements are immediately available.

First, any inconsistent set of literals is a violating set of any set R of rules. (It has no consistent superset.) But an inconsistent set of literals contributes nothing useful to $aux(R)$. The inconsistent set $\{c, {\sim}c\}$ produces only the pair $\{c \boxrightarrow c,\ {\sim}c \boxrightarrow {\sim}c\}$ in $aux(R)$. These are merely instances of MUST-ID. More generally, an inconsistent set $A \cup \{c, {\sim}c\}$ contributes the following rules to $aux(R)$:

$$right(\{A \wedge c \wedge {\sim}c \boxrightarrow \bot\}) = \begin{cases} A \wedge c \boxrightarrow c \\ A \wedge {\sim}c \boxrightarrow {\sim}c \\ (A - \{a\}) \wedge c \wedge {\sim}c \boxrightarrow \overline{a} \quad (\text{all } a \in A) \end{cases}$$

The first two rules are merely consequences of MUST-ID and MUST-SI. The others are entailed by $c \wedge {\sim}c \boxrightarrow \bot$ in KL_1 by MUST-SI and QUOD-LIBET. So if X is an inconsistent set of literals, then $right(\{X \boxrightarrow \bot\}) \subseteq kl_1(C)$: X contributes nothing to $out_2(R, A)$ and can be ignored.

Second, if X is a violating set of R and $X' \supseteq X$ then X' is also a violating set of R. Moreover, every rule in $right(\{X' \boxrightarrow \bot\})$ can be derived in KL_1 from

a rule in $right(\{X \boxdot\!\!\rightarrow \bot\})$. For suppose $X' = X \cup Y$, X and Y disjoint. Then the rules in $right(\{X' \boxdot\!\!\rightarrow \bot\})$ are of the following two forms:

$$right(\{X \cup Y \boxdot\!\!\rightarrow \bot\}) = \begin{cases} (X - \{a\}) \cup Y \boxdot\!\!\rightarrow \overline{a} & (\text{all } a \in X) \\ X \cup (Y - \{a\}) \boxdot\!\!\rightarrow \overline{a} & (\text{all } a \in Y) \end{cases}$$

Rules in the first group are derived by MUST-SI from $(X - \{a\}) \boxdot\!\!\rightarrow \overline{a}$. Rules in the second group are derived from $X \boxdot\!\!\rightarrow \bot$ by MUST-SI and QUOD-LIBET. So if $X' \supseteq X$ then $right(\{X' \boxdot\!\!\rightarrow \bot\}) \subseteq kl_1(right(\{X \boxdot\!\!\rightarrow \bot\}))$. This means that in the construction of $aux(R)$ it is enough to consider the *minimal* violating sets of R.

Definition 9 Let R be a set of rules. Define:

$$aux_m(R) \;=\; \{A - \{a\} \boxdot\!\!\rightarrow \overline{a} \mid A \text{ is a minimal consistent violating set of } R, a \in A\}$$

Proposition 10 *Let R be a set of rules and A a set of literals.*

$$aux_m(R) \subseteq aux(R) \subseteq kl_1(aux_m(R) \cup C)$$

and hence

$$out_2(R, A) = out_1(R \cup C \cup aux_m(R),\ A)$$

Proof In the preceding discussion. □

Note that if $R \cup C$ is strongly inconsistent then $\emptyset$ is the only minimal consistent violating set of R. In that case $aux_m(R) = \emptyset$ and $out_2(R, A) = out_1(R \cup C, A) = \emptyset$ for all sets A of literals. (The converse does not hold, as observed earlier.)

Example 12

$$R = \{\, \top \boxdot\!\!\rightarrow p \vee q,\ p \wedge q \boxdot\!\!\rightarrow \bot \,\}$$

$\{\sim p, \sim q\}$, $\{p, q\}$ are the only minimal consistent violating sets of R. (There are other violating sets, but they are either inconsistent or non-minimal.)

$$aux_m(R) \;=\; \left\{ \begin{matrix} \sim p \boxdot\!\!\rightarrow q & \quad p \boxdot\!\!\rightarrow \sim q \\ \sim q \boxdot\!\!\rightarrow p & \quad q \boxdot\!\!\rightarrow \sim p \end{matrix} \right\}$$

$$out_2(R, \emptyset) \;=\; out_1(R \cup C \cup aux_m(R),\ \emptyset) \;=\; \{\{p, \sim q\},\ \{q, \sim p\}\}$$

Example 13

$$R = \{\, p \wedge q \boxdot\!\!\rightarrow \bot,\ \sim p \wedge q \boxdot\!\!\rightarrow \bot \,\}$$

$\{q\}$, $\{p, q\}$, $\{\sim p, q\}$ are consistent violating sets of R. (There are others.) $\{q\}$ is the only minimal consistent violating set.

$$aux_m(R) \;=\; \{\, \top \boxdot\!\!\rightarrow \sim q \,\}$$

$$out_2(R, \emptyset) \;=\; out_1(R \cup C \cup aux_m(R),\ \emptyset) \;=\; \{\{\sim q\}\}$$

Example 14

$$R = \{\, p \boxrightarrow q,\ p \boxrightarrow r,\ q \wedge r \boxrightarrow \bot \,\}$$

$\{p\}$, $\{p,q\}$, $\{p,\sim q\}$, $\{p,r\}$, $\{p,\sim r\}$, $\{p,q,r\}$, $\{p,q,\sim r\}$, $\{p,\sim q,r\}$, $\{p,\sim q,\sim r\}$, $\{q,r\}$, $\{\sim p,q,r\}$ are consistent violating sets of R. (There are others.) $\{p\}$ and $\{q,r\}$ are the minimal violating sets.

$$\begin{aligned} aux_m(R) &= \{\, \top \boxrightarrow \sim p,\ q \boxrightarrow \sim r,\ r \boxrightarrow \sim q \,\} \\ out_2(R,\emptyset) &= out_1(R \cup C \cup aux_m(R),\ \emptyset) = \{\{\sim p\}\} \\ out_2(R,\{q\}) &= out_1(R \cup C \cup aux_m(R),\ \{q\}) = \{\{\sim p, q, \sim r\}\} \end{aligned}$$

Finally we confirm that $out_2(R,A)$ is well-defined for non-empty sets A of assumptions.

Proposition 11 *Let R be a set of rules and A a set of literals.*

$$out_2(R,A) = out_2(R \cup \{\top \boxrightarrow a \mid a \in A\},\ \emptyset)$$

Proof The result follows from the previous minimality result. It is enough to consider a singleton set of assumptions $A = \{a\}$. The general result follows by repeated application.

If $R \cup \{\top \boxrightarrow a\} \cup C$ is strongly inconsistent the result holds trivially. Suppose it is not strongly inconsistent. We need to show that:

$$out_1(R \cup \{\top \boxrightarrow a\} \cup C \cup aux(R \cup \{\top \boxrightarrow a\})) = out_1(R \cup \{\top \boxrightarrow a\} \cup C \cup aux(R))$$

We will show that $aux_m(R \cup \{\top \boxrightarrow a\}) = aux_m(R) \cup \{\top \boxrightarrow a\}$.

Clearly $\{\overline{a}\}$ and all violating sets of R are violating sets of $R \cup \{\top \boxrightarrow a\}$. Further, $right(\overline{a} \boxrightarrow \bot) = \{\top \boxrightarrow a\}$. So $aux(R) \cup \{\top \boxrightarrow a\} \subseteq aux(R \cup \{\top \boxrightarrow a\})$. Now (assuming $R \cup \{\top \boxrightarrow a\} \cup C$ is not strongly inconsistent) $\{\overline{a}\}$ is a minimal consistent violating set of $R \cup \{\top \boxrightarrow a\}$. If X is a minimal consistent violating set of R and $\overline{a} \in X$ then X is not a minimal consistent violating set of $R \cup \{\top \boxrightarrow a\}$; if $\overline{a} \notin X$ then X is a minimal consistent violating set of $R \cup \{\top \boxrightarrow a\}$. So $aux_m(R \cup \{\top \boxrightarrow a\}) = aux_m(R) \cup \{\top \boxrightarrow a\}$. □

4.4 An alternative characterisation of out_2

It is possible to construct alternative, equivalent characterisations of the auxiliary rules $aux(R)$. The following will be used in discussions of KL_3 to come and for completeness of KL_2.

Observation 1 *X is a violating set of R iff X is a violating set of rules $R_\bot \cup must_\bot(R)$ where $R_\bot$ is the set of constraint rules of the form $B \boxrightarrow \bot$ in R and $must_\bot(R)$ is the set of rules obtained by applying* MUST-⊥ *to the rules in R:*

$$must_\bot(R) = \{B \wedge \overline{c_1} \wedge \ldots \wedge \overline{c_k} \boxrightarrow \bot \mid (B \boxrightarrow c_1 \vee \ldots \vee c_k) \in R\}$$

Observation 2 *Let R_1 and R_2 be sets of rules. If X is a violating set of $R_1 \cup R_2$ then $X \subseteq X_1 \cup X_2$ for some X_1 and X_2 such that X_1 is a violating set of R_1 and X_2 is a violating set of R_2.*

The following is a derived rule of KL_2:

$$\text{REDUCE-}\bot \ \frac{B \wedge c \ \square\!\!\rightarrow \bot \quad B \wedge {\sim}c \ \square\!\!\rightarrow \bot}{B \ \square\!\!\rightarrow \bot}$$

Its derivation requires ∼-RIGHT and so it is an inference rule of KL_2 not of $KL_1(C)$. We can also give a semantic justification by appeal to violating sets. If $B \cup \{c\}$ and $B \cup \{{\sim}c\}$ are both violating sets of R then clearly so is B.

The following more general rule is also easily derived in KL_2:

$$\text{RESOLVE-}\bot \ \frac{A \wedge c \ \square\!\!\rightarrow \bot \quad B \wedge {\sim}c \ \square\!\!\rightarrow \bot}{A \cup B \ \square\!\!\rightarrow \bot}$$

Semantically, if $A \cup \{c\}$ and $B \cup \{{\sim}c\}$ are both violating sets of R then so is $A \cup B$. (We cannot extend consistently by either c or ${\sim}c$.) These rules will feature prominently in KL_3 and we will present their derivation there.

We can reformulate RESOLVE-⊥ as a rule applying to violating sets, exactly as stated in the semantic argument. We will call that V-RESOLVE.

Definition 10 Let Γ be a set of sets of literals, where each component set of literals is a violating set. Then V-RESOLVE$(\Gamma) = \Gamma \cup \{A \cup B \mid A \cup \{a\} \in \Gamma, B \cup \{\overline{a}\} \in \Gamma\}$.

Now we shall use V-RESOLVE to provide an alternative version of *aux*, called aux_m which provides the minimal set of auxiliary rules that are needed to derive all the consequences we want.

Definition 11

$$v(R) = \{B \cup \overline{C} \mid B \ \square\!\!\rightarrow C \in R, \ B \cup \overline{C} \text{ consistent}\}$$

(As observed earlier, that also covers the case of constraint rules, where $C = \emptyset$.) $v(R)$ are all consistent violating sets of R (though there may be others). Now define $v^*(R)$ as the closure of $v(R)$ under V-RESOLVE. Let $v^*_m(R)$ be the set of the minimal elements of $v^*(R)$. Assuming the construction is complete (see below) we define aux_m as follows:

$$aux_m(R) \ = \ \{A - \{a\} \ \square\!\!\rightarrow \overline{a} \mid A \in v^*_m(R), \ a \in A\}$$

We could reformulate RESOLVE-⊥ so that it does not generate rules with inconsistent bodies and V-RESOLVE so that inconsistent sets are discarded. We could also make the computation of $v^*(R)$ more efficient by discarding non-minimal elements as soon as they are constructed during the computation of the closure $v^*(R)$. These are details.

Example 15

$$R = \{\, p \wedge q \boxminus\!\!\rightarrow \bot,\ p \wedge {\sim}q \boxminus\!\!\rightarrow \bot \,\}$$

$$\begin{aligned} v(R) &= \{\{p,q\},\{p,{\sim}q\}\} \\ v^*(R) &= \{\{p,q\},\{p,{\sim}q\},\{p\}\} \\ v^*_m(R) &= \{\{p\}\} \\ aux_m(R) &= \{\, \top \boxminus\!\!\rightarrow {\sim}p \,\} \end{aligned}$$

Example 16

$$R = \{\, p \boxminus\!\!\rightarrow q,\ p \boxminus\!\!\rightarrow r,\ q \wedge r \boxminus\!\!\rightarrow \bot \,\}$$

$$\begin{aligned} v(R) &= \{\{p,{\sim}q\},\{p,{\sim}r\},\{q,r\}\} \\ v^*(R) &= v(R) \cup \{\{p,r\},\{p,q\},\{p\}\} \\ v^*_m(R) &= \{\{p\},\{q,r\}\} \\ aux_m(R) &= \{\, \top \boxminus\!\!\rightarrow {\sim}p,\ q \boxminus\!\!\rightarrow {\sim}r,\ r \boxminus\!\!\rightarrow {\sim}q \,\} \end{aligned}$$

Example 17

$$R = \{\, p \wedge q \boxminus\!\!\rightarrow r,\ p \wedge {\sim}q \boxminus\!\!\rightarrow r,\ p \wedge r \boxminus\!\!\rightarrow \bot \,\}$$

$$\begin{aligned} v(R) &= \{\{p,q,{\sim}r\},\{p,{\sim}q,{\sim}r\},\{p,r\}\} \\ v^*(R) &= v(R) \cup \{\{p,{\sim}r\},\{p,q\},\{p,{\sim}q\},\{p\}\} \\ v^*_m(R) &= \{\{p\}\} \\ aux_m(R) &= \{\, \top \boxminus\!\!\rightarrow {\sim}p \,\} \end{aligned}$$

Now we can define an alternative set of auxiliary rules $aux_e(R)$ to be used in $out_2(R, A)$ that will be useful in establishing completeness and in KL_3.

Definition 12 For all rule sets R, let:

$$aux_e(R) = \{A - \{a\} \boxminus\!\!\rightarrow \overline{a} \mid A \in v^*(R),\ a \in A\}$$

In order to use $aux_e(R)$ in the computation of $out_2(R, A)$, and to preserve semantic entailment $\models_{\mathrm{KL}_2}$, it is not necessary that $aux_e(R)$ generates all elements of $aux(R)$ – only that it generates at least all minimal elements $aux_m(R)$ of $aux(R)$ and nothing that is not in $aux(R)$.

Proposition 12 *Let R be a set of rules and X a set of literals. If X is a violating set of R and $X \notin v^*(R)$ then either X is inconsistent or there exists $X' \subset X$ such that $X' \in v*(R)$.*

Proof By induction on the number of rules in R, and Observation 2. □

This does not say that all elements of $v^*(R)$ are consistent or minimal, but only that all minimal consistent violating sets of R are elements of $v^*(R)$, which is all we need.

Clearly $aux_m(R) \subseteq aux_e(R)$. Further, since the set of all violating sets of R is closed under V-RESOLVE, $aux_e(R) \subseteq aux(R)$. We also have (Proposition 10) $aux(R) \subseteq kl_1(aux_m(R) \cup C)$. Putting these observations together gives the following.

Proposition 13 *Let R be a set of rules.*

$$aux_m(R) \subseteq aux_e(R) \subseteq aux(R) \subseteq kl_1(aux_m(R) \cup C)$$

and hence

$$kl_1(R \cup C \cup aux(R)) \;=\; kl_1(R \cup C \cup aux_e(R))$$

Now we shall provide an alternative characterisation of $aux_e(R)$ in terms of inference rules of KL_2.

Definition 13 Let $right^+(R)$ denote the results of applying inference rule ~-RIGHT to rules R keeping only those rules whose bodies are consistent:

$$right^+(R) \;=\; \{A - \{a\} \boxdot\!\!\rightarrow \overline{a} \mid A \boxdot\!\!\rightarrow \bot \in R,\ A \text{ is consistent},\ a \in A\}$$

Let $resolve^*_\bot(R)$ denote the closure of rules R under derived rule RESOLVE-$\bot$, i.e., the closure of R under:

$$resolve_\bot(R) = \{B \cup B' \boxdot\!\!\rightarrow \bot \mid B \wedge c \boxdot\!\!\rightarrow \bot \in R, B \wedge \overline{c} \boxdot\!\!\rightarrow \bot \in R\}$$

Let $must_\bot(R)$ denote the application of derived rule MUST-$\bot$ to R:

$$must_\bot(R) = \{B \cup \overline{C} \boxdot\!\!\rightarrow \bot \mid B \boxdot\!\!\rightarrow C \in R\}$$

Now we can provide an alternative characterisation of aux_e:

Proposition 14 *Let R be a set of rules.*

$$aux_e(R) = right^+(resolve^*_\bot(R \cup must_\bot(R)))$$

Proof This follows from the definitions. $must_\bot(R)$ is the set of constraint rules implied by rules of the form $B \boxdot\!\!\rightarrow C$ ($C \neq \emptyset$) in R. R may also contain constraint rules of the form $B \boxdot\!\!\rightarrow \bot$. So (by definition) $X \in v(R)$ when $X \boxdot\!\!\rightarrow \bot$ is a rule in $R \cup must_\bot(R)$ and X is consistent. $X \in v^*(R)$ when $X \boxdot\!\!\rightarrow \bot$ is a rule in $resolve^*_\bot(R \cup must_\bot(R))$ and X is consistent. So $aux_e(R) = right^+(resolve^*_\bot(R \cup must_\bot(R)))$. □

Now this is going to be used in establishing completeness, because all the inference rules used in the construction of $aux_e(R)$ are (derived) inference rules of KL_2.

4.5 Soundness and completeness

The inference rules of KL_2 are those of KL_1 together with ~-LEFT and ~-RIGHT. We write $deriv_2(R)$ to denote the set of rules that can be derived from the set R of rules by repeated application of the inference rules of KL_2. We write $R \vdash_{KL_2} r$ if $r \in deriv_2(R)$.

Proposition 15 (Soundness of KL_2) *For all sets R of rules:*

$$deriv_2(R) \subseteq kl_2(R)$$

Proof We need to show soundness of the inference rules of KL_1, ~-LEFT and ~-RIGHT with respect to semantic entailment in KL_2. Since $kl_2(R) = kl_1(R \cup C \cup aux(R))$ (Proposition 6) and KL_1 is sound with respect to kl_1, the soundness of KL_1 inference rules is immediate. Soundness of ~-LEFT is just $C \subseteq kl_2(R)$ which also follows trivially.

It remains to show that ~-RIGHT is sound: if $r \in right(R)$ then $R \models_{KL_2} r$, or more generally, that $right(R) \subseteq kl_2(R)$. We want to show:

$$right(R) \subseteq kl_1(R \cup C \cup aux(R))$$

We will show that $right(R) \subseteq aux(R)$ (which implies the above). In full: if $r \in right(R)$ then r is a rule of the form $B \boxminus\!\!\!\rightarrow \bar{c}$ where $B \wedge c \boxminus\!\!\!\rightarrow \bot$ is a rule in R. In that case $B \cup \{c\}$ is a (not necessarily minimal or consistent) violating set of R, and $aux(R)$ contains the rule $B \boxminus\!\!\!\rightarrow \bar{c}$. □

We would expect that if KL_1 is complete with respect to out_1 then KL_2 is complete with respect to out_2. That is indeed the case.

Proposition 16 (Completeness of KL_2) *If KL_1 is complete with respect to out_1 then KL_2 is complete with respect to out_2. That is: if, for all sets R of rules $kl_1(R) \subseteq deriv_1(R)$ then, for all sets R of rules $kl_2(R) \subseteq deriv_2(R)$.*

Proof

$$\begin{aligned}
r \in kl_2(R) &\Rightarrow r \in kl_1(R \cup C \cup aux(R)) \\
&\Rightarrow r \in kl_1(R \cup C \cup aux_e(R)) && \text{(Proposition 13)} \\
&\Rightarrow r \in deriv_1(R \cup C \cup aux_e(R)) && \text{(completeness of } KL_1\text{)} \\
&\Rightarrow r \in deriv_2(R)
\end{aligned}$$

The final step is because all the inference rules used in the construction of $aux_e(R)$ in Proposition 14 are inference rules of KL_2. □

Proposition 17 (Decomposition of KL_2) *If KL_1 is complete with respect to out_1 then*

$$deriv_2(R) = deriv_1(R \cup C \cup aux_e(R))$$

Proof Right-in-left inclusion is noted in the proof of Proposition 16. The other inclusion is similar:

$$\begin{aligned} r \in \mathit{deriv}_2(R) &\Rightarrow r \in \mathit{kl}_2(R) && \text{(soundness of KL}_2\text{)} \\ &\Rightarrow r \in \mathit{kl}_1(R \cup C \cup \mathit{aux}(R)) \\ &\Rightarrow r \in \mathit{kl}_1(R \cup C \cup \mathit{aux}_e(R)) && \text{(Proposition 13)} \\ &\Rightarrow r \in \mathit{deriv}_1(R \cup C \cup \mathit{aux}(R)) && \text{(completeness of KL}_1\text{)} \end{aligned}$$

□

4.6 Conservative extension

We have established that:

$$\begin{aligned} \mathit{kl}_2(R) &= \mathit{kl}_1(R \cup C \cup \mathit{aux}(R)) \\ \mathit{deriv}_2(R) &= \mathit{deriv}_1(R \cup C \cup \mathit{aux}_e(R)) \end{aligned}$$

One can see that KL_2 is a conservative extension of KL_1, both semantically and syntactically.

Proposition 18 (Conservative extension) *KL_2 is a conservative extension of KL_1: If R is a set of rules containing no negative literals, and rule r also contains no negative literals, then $r \in \mathit{kl}_2(R)$ iff $r \in \mathit{kl}_1(R)$, and $r \in \mathit{deriv}_2(R)$ iff $r \in \mathit{deriv}_1(R)$.*

We can see this claim is true by looking at the $\mathit{aux}(R)$ construction: if R contains no negative literals, then all violating sets of R are sets of atoms. All the rules in $\mathit{aux}(R)$ are therefore rules with singleton heads where the head is a negative literal and the body contains only positive atoms, i.e., rules of the form $B \mathrel{\square\!\!\rightarrow} {\sim}c$ where c is an atom and B is a set (possibly empty) of atoms. Any rule r containing only positive atoms can only be derived from $R \cup \mathit{aux}(R)$ (syntactically or semantically) if it can be derived from the rules R. The constraint rules C have no effect if neither R nor the entailed rule r contain negative literals.

Indeed, $\mathrm{KL}_1(C)$ is a conservative extension of KL_1 and KL_2 is a conservative extension of $\mathrm{KL}_1(C)$. $\mathit{kl}_1(R \cup C)$ is a conservative extension of $\mathit{kl}_1(R)$ and $\mathit{kl}_2(R)$ is a conservative extension of $\mathit{kl}_1(R \cup C)$.

KL_2 can also be seen as a conservative extension of KL_1 in the following rather different sense. Given a set X of literals, let X^+ be the largest subset of X containing only positive literals. In other words, let X^+ be the set of atoms obtained by removing all negative literals from X. If Δ is a set of sets of literals, let $\Delta^+ = \{X^+ \mid X \in \Delta\}$.

Proposition 19 *Let R be a set of rules and A a set of assumptions, both containing no negative literals. Then:*

$$\mathit{out}_2(R, A)^+ = \mathit{out}_1(R, A)$$

Proof We can see this again by looking at the rules in $aux(R)$: if R contains no negative literals, all rules in $aux(R)$ are of the form $B \boxdot\!\!\rightarrow \sim c$ where c and all of B are atoms. These rules have no effect on the outcomes computed from R except possibly to add negative literals. This is clear if we look at the translation to definite logic programs: every definite program D in the encoding $def(R \cup C \cup aux(R))$ has the form $D_R \cup D_{aux}$ where $D_R \in def(R)$ and $def(aux(R)) = \{D_{aux}\}$ (all rules in $aux(R)$ are $\boxdot\!\!\rightarrow$ rules with singleton heads and so there is a single definite program encoding $aux(R)$). Moreover, since none of the heads of clauses in D_{aux} appear in D_R the least model $\mathrm{M}(D_R \cup D_{aux}, A) = \mathrm{M}(D_R, \mathrm{M}(D_{aux}, A))$ (indeed that is true whether the set A of assumptions contains negative literals or not). If A contains no negative literals, then this least model also satisfies the constraints C. □

In other words: out_2 does not add or remove from the set of solutions to out_1 – all it does is possibly add some negative literals to the existing solutions.

Example 18 Let $R = \{p \boxdot\!\!\rightarrow q,\ p \boxdot\!\!\rightarrow r,\ q \wedge r \boxdot\!\!\rightarrow \bot\}$. Then $out_2(R) = \{\{\sim p\}\}$ and $out_1(R) = \{\emptyset\}$.

4.7 Entailments

Some examples of entailments and non-entailments are given in Table 2. Note that the rule corresponding to the law of excluded middle ($\top \boxdot\!\!\rightarrow p \vee \sim p$) is *not* a theorem of KL_2.

Table 2: Some entailments and non-entailments in KL_2

Entailments	**Non-entailments**
$\{\} \models_{\mathrm{KL}_2} p \wedge \sim p \boxdot\!\!\rightarrow q$	$\{\} \not\models_{\mathrm{KL}_2} \top \boxdot\!\!\rightarrow p \vee \sim p$
$p \boxdot\!\!\rightarrow \bot \models_{\mathrm{KL}_2} \top \boxdot\!\!\rightarrow \sim p$	$p \wedge q \boxdot\!\!\rightarrow \bot \not\models_{\mathrm{KL}_2} \top \boxdot\!\!\rightarrow \sim p \vee \sim q$
$p \boxdot\!\!\rightarrow q \models_{\mathrm{KL}_2} p \boxdot\!\!\rightarrow q \vee \sim q$	$p \boxdot\!\!\rightarrow q \not\models_{\mathrm{KL}_2} p \boxdot\!\!\rightarrow q \vee r$
$p \boxdot\!\!\rightarrow q \models_{\mathrm{KL}_2} \sim q \boxdot\!\!\rightarrow \sim p$	$p \boxdot\!\!\rightarrow q \not\models_{\mathrm{KL}_2} \sim p \boxdot\!\!\rightarrow \sim q$
$\sim q \boxdot\!\!\rightarrow \sim p \models_{\mathrm{KL}_2} p \boxdot\!\!\rightarrow q$	$\sim p \boxdot\!\!\rightarrow \sim q \not\models_{\mathrm{KL}_2} p \boxdot\!\!\rightarrow q$
$\top \boxdot\!\!\rightarrow \sim p \vee q \models_{\mathrm{KL}_2} p \boxdot\!\!\rightarrow q$	$p \boxdot\!\!\rightarrow q \not\models_{\mathrm{KL}_2} \top \boxdot\!\!\rightarrow \sim p \vee q$

4.8 Concluding remarks on negation

The treatment of negation in KL_2 derives from two starting assumptions: that complementary literals c and $\sim c$ are mutually incompatible (∼-LEFT), and that the negation of c is the most general proposition that is incompatible with c (∼-RIGHT). These two assumptions embedded in KL_1 produce a form of negation in which the rules $B \wedge c \boxdot\!\!\rightarrow \bot$ and $B \boxdot\!\!\rightarrow \overline{c}$ turn out to be equivalent[33]. It is

[33] This equivalence only holds at the propositional level. We shall see in Section 5.1 below that the two rules are *not* equivalent when one of them contains existentially quantified variables.

possible to devise some more elaborate technical constructions which weaken this equivalence, such that the rule $B \boxdot\!\!\rightarrow \overline{c}$ entails $B \wedge c \boxdot\!\!\rightarrow \bot$ but not the other way round. We have not presented any such alternatives here. The technical constructions are not difficult but we have not found support for them in Kant's writings.

5 KL_3: extending KL_2 with variables and quantifiers

KL_3 extends KL_2 by adding quantified rules to KL_2, including rules in which the head may have existentially quantified variables. In KL_3, an atom has internal structure; it is composed of a predicate and a list of terms.

Given a set $\mathcal{P}$ of predicate symbols with associated arities[34]:

$$\mathcal{P}^{+/-} = \mathcal{P} \cup \{\sim p \mid p \in \mathcal{P}\}$$

The negation of a predicate p means "un-p". For example, $\sim$*clear* means "unclear". The formula $p(x) \boxdot\!\!\rightarrow \sim q(x)$ does *not* mean "if $p(x)$ then do not subsume x under q!". Rather, it means: "if $p(x)$ then do subsume x under un-q!".

Given a set $\mathcal{P}^{+/-}$ of predicate symbols, a set $\mathcal{K}$ of constants, and a set $\mathcal{X}$ of variables, the set $\mathcal{L}_K$ of **ground literals** is:

$$\mathcal{L}_K ::= \{p(k_1, \ldots, k_n) \mid p \in \mathcal{P}^{+/-},\ k_i \in \mathcal{K},\ \mathsf{arity}(p) = n\}$$

The set $\mathcal{L}_X$ of **unground literals** is:

$$\mathcal{L}_X ::= \{p(x_1, \ldots, x_n) \mid p \in \mathcal{P}^{+/-},\ x_i \in \mathcal{X},\ \mathsf{arity}(p) = n\}$$

The set $\mathcal{L}$ of literals is:

$$\mathcal{L} ::= \{p(t_1, \ldots, t_n) \mid p \in \mathcal{P}^{+/-},\ t_i \in \mathcal{K} \cup \mathcal{X},\ \mathsf{arity}(p) = n\}$$

Note that predicates of arity 0 are allowed. A literal of arity 0 is both a grounded literal and an ungrounded literal.

In what follows, constants and variables are written in lower case. Constants are a, b, c, while variables are x, y, z, possibly with subscripts. To avoid cluttering the syntax, we take it to be obvious from context whether a, b, c are to be read as constants or as ranging over literals.

Note that both $\mathcal{L}_K$ and $\mathcal{L}_X$ are proper subsets of $\mathcal{L}$, and there are literals in $\mathcal{L}$ that are not in $\mathcal{L}_K$ nor $\mathcal{L}_X$: any literal that contains a mixture of variables and constants is in neither $\mathcal{L}_K$ nor $\mathcal{L}_X$. $p(x, k)$ is not in $\mathcal{L}_K$ nor in $\mathcal{L}_X$.

In KL_3, *rules are made up entirely of unground literals* from $\mathcal{L}_X$. No constants are allowed in any of the literals in any of the rules. This is essential. Since rules are intended to be public and shareable between agents, while intuitions are private mental objects, rules must not contain constants (intuitions) or they would not be public (see Section 2.2).

[34] Some commentators believe Kant's logic only allowed monadic predicates. But [1] and [2] argue convincingly that Kant always had n-ary predicates in mind.

There are two forms of rule in KL_3, as in KL_1 and KL_2 but with B and C ranging over sets of unground literals from $\mathcal{L}_X$:

$$\mathcal{R} ::= B \Box\!\!\rightarrow C \mid B \Diamond\!\!\rightarrow C$$

Variables that appear in both the body and the head of a rule are read as universally quantified. For example, $p(x) \Box\!\!\rightarrow q(x)$ means: "for any x, if you perform $p(x)$, then you must perform $q(x)$!". Variables that appear in the head but not in the body are existentially quantified[35]. For example, $p(x) \Diamond\!\!\rightarrow q(x, y)$ means: "for any x, if you perform $p(x)$, then you may construct a y and perform $q(x, y)$!". $p(x) \Box\!\!\rightarrow q(x, y)$ means: "for any x, if you perform $p(x)$, then you must construct a y and perform $q(x, y)$!". To emphasise this reading, we write such rules with explicit existential quantifiers in the head, as in e.g. $p(x) \Diamond\!\!\rightarrow \exists y\, q(x, y)$ and $p(x) \Box\!\!\rightarrow \exists y\, q(x, y)$.

A rule such as $p(x) \Box\!\!\rightarrow q(x, y) \vee r(x, y)$ where there is a shared variable y in the head can be read either as $p(x) \Box\!\!\rightarrow \exists y\, (q(x, y) \vee r(x, y))$ or (equivalently) as $p(x) \Box\!\!\rightarrow \exists y\, q(x, y) \vee \exists y\, r(x, y)$. Notice that the latter is also equivalent to $p(x) \Box\!\!\rightarrow \exists y\, q(x, y) \vee \exists z\, r(x, z)$, and therefore to the rule $p(x) \Box\!\!\rightarrow q(x, y) \vee r(x, z)$ without explicit quantifiers.

As explained below, however, for simplicity of presentation and for practical reasons we will restrict the language so that existential rules have only singleton heads. This does not restrict the expressive power of the language.

5.1 Preliminaries

Where θ is a substitution (an assignment of variables and/or constants to variables) and c is a literal, the expression $c.\theta$ denotes the application of θ to c. Where C is a set of literals $C.\theta = \{c.\theta \mid c \in C\}$. A substitution θ is ground when all variables in θ are assigned to constants. Where C is a set of unground literals and $C.\theta$ are ground literals we say that $C.\theta$ is a ground instantiation of C.

Example Suppose $\theta = \{x/a\}$ and $\theta' = \{y/b\}$. Then

$$p(x).\theta = p(a)$$
$$q(x, y).\theta.\theta' = q(a, y).\theta' = q(a, b)$$

Suppose $\theta = \{x/a, y/b, z/c\}$ and $\theta' = \{\}$ (the identity substitution). Then

$$p(x).\theta = p(a)$$
$$q(x, y).\theta.\theta' = q(a, b).\theta' = q(a, b)$$

[35] Rules with existentials in the head are common in geometric logic [14] (also known as coherent logic [5,6]), existential datalog [11], and in agent languages [35]. See Section 5.5 for further comparison.

Definition 14 A set X of ground literals satisfies a set of rules R, written $X \models R$, when X satisfies every rule in R. X satisfies a rule r, written $X \models r$, when:

$X \models B \boxdot\!\!\rightarrow C$	if for every ground instantiation $B.\theta$ of B, if $B.\theta \subseteq X$ then there exists a ground instantiation $C.\theta.\theta'$ of $C.\theta$ such that $C.\theta.\theta' \cap X \neq \emptyset$
$X \models B \diamond\!\!\rightarrow C$	always

Note in the above that if there are no existential variables in the head C of a rule $B \boxdot\!\!\rightarrow C$ then $C.\theta$ is ground and θ' is the identity substitution. If there are existentially quantified variables in C and θ instantiates all the variables in C to constants (i.e., if $C.\theta$ is already a ground instantiation of C) then θ' is the identity substitution.

Rules in KL_3 are quantified and unground. Leaving aside rules with existential heads, it is clear that *formally* all ground instances of KL_3 rules are – syntactically and semantically – rules of KL_2 where the positive ground literals of KL_3 are treated as positive literals (atoms) of KL_2, and negative ground literals of KL_3 as negative literals of KL_2, i.e., as atoms prefixed by the negation operator. We can see that, for rules without existential heads:

$X \models B \boxdot\!\!\rightarrow C$	if $X \models B.\theta \boxdot\!\!\rightarrow C.\theta$ for every ground instantiation $B.\theta$ of B
$X \models B \diamond\!\!\rightarrow C$	always

Universally quantified rules without existentially quantified heads behave exactly in KL_3, syntactically and semantically, as the set of all their ground instances in KL_2.

Rules with existentially quantified heads however are a *different kind of rule* and have to be treated specially. Consider the very simplest example:

$$p \boxdot\!\!\rightarrow \exists x\, q(x)$$

At first sight it might seem that this rule cannot be violated, that there is (apparently) no consistent violating set because we can always extend a (consistent) set of ground literals by finding a new candidate $q(k_i)$ atom.

But that is not so. The set $\{p, {\sim}q(a_0), {\sim}q(a_1), \dots\}$ (infinitely many ${\sim}q(a_i)$ literals) is a violating set, as are all of its supersets. Ordinarily, in the semantics adopted for negation in KL_2, if X is a violating set of rule set R then R entails the rule $X \boxdot\!\!\rightarrow \bot$. That does not work here: X in this example would represent an infinite conjunction, which is not well-formed. Put another way, the derived inference rule MUST-$\bot$ which we rely on in the construction of $aux_e(R)$ in KL_2, would look as follows:

$$\frac{p \boxdot\!\!\rightarrow \exists x\, q(x)}{p \wedge {\sim}q(x) \boxdot\!\!\rightarrow \bot}$$

That rule does not hold for existential rules. The quantification is wrong. To make it valid we would need

$$\frac{p \boxdot\!\!\rightarrow \exists x\, q(x)}{p \wedge {\sim}\exists x\, q(x) \boxdot\!\!\rightarrow \bot}$$

but the consequent is not an allowed rule form in KL_3.

What about $\bot$-RIGHT? Could the following be valid?

$$\frac{p \wedge q(x) \boxdot\!\!\rightarrow \bot}{p \boxdot\!\!\rightarrow \exists x \sim q(x)}$$

Clearly not. "You must not perform p and $q(x)$ for any x!" should not imply "if you perform p you must also construct an x and perform $\sim q(x)$!". For KL_3 we will need a restricted form of $\bot$-RIGHT, as discussed in the next section. For example, the following inference is valid:

$$\frac{p \wedge \sim q(x) \boxdot\!\!\rightarrow \bot}{\sim q(x) \boxdot\!\!\rightarrow \sim p}$$

Further, notice that the following rules

$$\begin{cases} \top \boxdot\!\!\rightarrow \exists x\, q(x) \\ q(x) \boxdot\!\!\rightarrow \bot \end{cases}$$

are strongly inconsistent (with the obvious definition). And that the following pair

$$\begin{cases} p \boxdot\!\!\rightarrow \exists x\, q(x) \\ q(x) \boxdot\!\!\rightarrow \bot \end{cases}$$

is weakly inconsistent and has a violating set $\{p\}$.

Now this is key, because we will want to construct a set $aux_e^q(R)$ of auxiliary rules for KL_3 in analogy to the construction of $aux_e(R)$ in KL_2. $aux_e(R)$ employs a combination of MUST-$\bot$, to derive constraint rules from non-constraint rules, and then RESOLVE-$\bot$ to process constraint rules. That is not available here – we do not have MUST-$\bot$ for existential rules. For existential rules we need (the general form of) the following inference rule, a generalisation of the example above:

$$\frac{A \boxdot\!\!\rightarrow \exists x\, q(x) \quad B \wedge q(x) \boxdot\!\!\rightarrow \bot}{A \cup B \boxdot\!\!\rightarrow \bot}$$

In the next section we will call the general form of the above inference rule EXISTS-$\bot$. In KL_2 its propositional analogue can be derived from MUST-$\bot$ followed by an application of RESOLVE-$\bot$. In KL_3 it can be given a semantic justification in terms of violation sets, as sketched above for the example. It is also derivable from the inference rules for KL_3 to be presented in the next section – however, as we show there, the derivation imposes certain restrictions on variables that limit its applicability in KL_3. Similarly, we are also going to need the KL_3 analogue of RESOLVE-$\bot$; its derivation likewise will impose certain restrictions in order to deal correctly with quantifiers.

For ease of exposition, we restrict attention to the special case of existential rules with singleton heads. Note that this restriction causes no loss of expressive power: we can express rules with existentially quantified disjunctive heads by introducing auxiliary predicates if necessary.

In summary we have:

- universally quantified rules without existentially quantified heads; they have exactly the same meaning – the same semantics and inference rules – as sets of all their ground instances in KL_2;
- inference rules for converting existential rules to constraint rules, which we can justify by appeal to violation sets, and which are derivable from the inference rules for KL_3 to be presented in the next section. The inference rule EXISTS-$\bot$ for existential rules with singleton heads is simple.

5.2 Inference Rules

The inference rules for KL_3 are provided in Figure 5. As explained above, for simplicity we deal only with the case of universally quantified rules and existential rules with singleton heads. In Figure 5, a, b, c range over unground literals, and A, B, C, A', B', C' range over sets of unground literals. The inference rules are of two types: those that are valid for all rules, and those that are valid only for universally quantified rules without existential variables in the head. In the figure they are distinguished by specifying restrictions on variables.

SUB-1 and SUB-2 are specific to KL_3. They allow the uniform replacement of variables by variables, enabling, for example, the inference from $p(x) \boxdot\!\!\rightarrow q(x)$ to $p(y) \boxdot\!\!\rightarrow q(y)$. In SUB-1 and SUB-2, the substitution θ must be injective on the existential variables (the variables in $B - A$). Without this restriction, they would allow the inference from $p(x) \boxdot\!\!\rightarrow \exists y\, q(x, y)$ to $p(x) \boxdot\!\!\rightarrow q(x, x)$, which is invalid.

In MUST-SI and MAY-SI, the new literals in $A' - A$ must not bind any of the existential variables in $B - A$. Without this restriction, we would be able to infer from $p(x) \boxdot\!\!\rightarrow \exists y\, q(x, y)$ to $p(x) \wedge r(y) \boxdot\!\!\rightarrow q(x, y)$, which is invalid.

In $\sim$-RIGHT, we insist that $var(c) \subseteq var(B)$. Without this restriction, we would be able to infer wrongly from $p(x) \wedge q(x, y) \boxdot\!\!\rightarrow \bot$ to $p(x) \boxdot\!\!\rightarrow \exists y \sim q(x, y)$.

Figure 6 shows three derived inference rules. They will be used, as in KL_2, in the construction of auxiliary rules $aux^q_e(R)$ used in the definition of the *out* function for KL_3.

MUST-$\bot$ was used in KL_2. It is valid for universally quantified rules without existentially quantified heads but not for rules with existentially quantified heads. Its derivation is presented below in order to show how the restrictions on variables are inherited from MUST-SI. For brevity we only show the derivation for the special case of a rule with singleton head. The derivation of the general form is easily reconstructed.

$$\dfrac{\dfrac{\dfrac{B \boxdot\!\!\rightarrow c}{B \wedge \overline{c} \boxdot\!\!\rightarrow c}\ \text{MUST-SI}\ \ var(\overline{c}) \subseteq var(B) \qquad \dfrac{\dfrac{}{\overline{c} \wedge c \boxdot\!\!\rightarrow \bot}\ \sim\text{-LEFT}}{B \wedge \overline{c} \wedge c \boxdot\!\!\rightarrow \bot}\ \text{MUST-SI}}{B \wedge \overline{c} \boxdot\!\!\rightarrow \bot}\ \text{MUST-TRANS}$$

EXISTS-$\bot$, also discussed informally in the previous section, gives the conditions under which we can derive a universally quantified constraint rule from an existential rule. We present only the version for existential rules with

$$\text{SUB-1}\ \frac{A \Box\!\!\rightarrow B}{A.\theta \Box\!\!\rightarrow B.\theta} \quad \text{when } \theta \text{ injective on } var(B - A)$$

$$\text{SUB-2}\ \frac{A \Diamond\!\!\rightarrow B}{A.\theta \Diamond\!\!\rightarrow B.\theta} \quad \text{when } \theta \text{ injective on } var(B - A)$$

$$\text{MUST-ID}\ \frac{-}{A \Box\!\!\rightarrow A}\ A \neq \emptyset \qquad \text{MUST-UNION}\ \frac{A \Box\!\!\rightarrow B \quad A \Diamond\!\!\rightarrow C}{A \Box\!\!\rightarrow B \cup C}$$

$$\text{MUST-SI}\ \frac{A \Box\!\!\rightarrow B}{A' \Box\!\!\rightarrow B} \quad A \subset A',\ var(A' - A) \cap var(B - A) = \emptyset$$

$$\text{MUST-TRANS}\ \frac{A \Box\!\!\rightarrow b_1 \vee \ldots \vee b_n \quad A \wedge b_1 \Box\!\!\rightarrow C \quad \ldots \quad A \wedge b_n \Box\!\!\rightarrow C}{A \Box\!\!\rightarrow C}$$

$$\text{QUOD-LIBET}\ \frac{A \Box\!\!\rightarrow \bot}{A \Box\!\!\rightarrow B}$$

$$\text{MAY-ID}\ \frac{-}{A \Diamond\!\!\rightarrow A} \qquad \text{MAY-UNION}\ \frac{A \Diamond\!\!\rightarrow B \quad A \Diamond\!\!\rightarrow C}{A \Diamond\!\!\rightarrow B \cup C}$$

$$\text{MAY-SI}\ \frac{A \Diamond\!\!\rightarrow B}{A' \Diamond\!\!\rightarrow B} \quad A \subset A',\ var(A' - A) \cap var(B - A) = \emptyset$$

$$\text{MAY-TRANS}\ \frac{A \Diamond\!\!\rightarrow b_1 \vee \ldots \vee b_n \quad A \wedge b_1 \Diamond\!\!\rightarrow C \quad \ldots \quad A \wedge b_n \Diamond\!\!\rightarrow C}{A \Diamond\!\!\rightarrow C}$$

if for every $c \in C$, $A \wedge c \Box\!\!\rightarrow b_i$ for some $b_i \in \{b_1, \ldots, b_n\}$

$$\text{MAY-SO}\ \frac{A \Diamond\!\!\rightarrow B}{A \Diamond\!\!\rightarrow B'} \quad B' \subset B$$

$$\text{MAY-MUST}\ \frac{A \Box\!\!\rightarrow B}{A \Diamond\!\!\rightarrow B} \qquad \text{MAY-FALSUM}\ \frac{A \wedge b \Box\!\!\rightarrow \bot}{A \Diamond\!\!\rightarrow b}$$

$$\sim\text{-LEFT}\ \frac{-}{c \wedge \sim c \Box\!\!\rightarrow \bot} \qquad \sim\text{-RIGHT}\ \frac{A \wedge b \Box\!\!\rightarrow \bot}{A \Box\!\!\rightarrow \overline{b}} \quad var(b) \subseteq var(A)$$

Fig. 5: Inference rules for KL_3

singleton heads. The rule can be justified semantically, by reference to violation sets, and also derived from the inference rules in Figure 5: the derivation uses MUST-SI and MUST-TRANS and for this reason EXISTS-$\bot$ inherits restrictions on variables from MUST-SI. Notice in particular that none of the existentially quantified variables in $A \Box\!\!\rightarrow c$ may appear in the literals B of $B \wedge c \Box\!\!\rightarrow \bot$.

$$\text{MUST-}\bot\ \frac{A \boxdot\!\!\rightarrow C}{A \cup \overline{C} \boxdot\!\!\rightarrow \bot} \quad var(C) \subseteq var(A)$$

$$\text{EXISTS-}\bot\ \frac{A \boxdot\!\!\rightarrow c \quad B \wedge c \boxdot\!\!\rightarrow \bot}{A \cup B \boxdot\!\!\rightarrow \bot} \quad (var(c) - var(A)) \cap var(B) = \emptyset$$

$$\text{RESOLVE-}\bot\ \frac{A \wedge c \boxdot\!\!\rightarrow \bot \quad B \wedge \overline{c} \boxdot\!\!\rightarrow \bot}{A \cup B \boxdot\!\!\rightarrow \bot} \quad var(c) \subseteq var(A) \text{ or } var(c) \subseteq var(B)$$

Fig. 6: Three derived inference rules of KL_3

$$\text{MUST-TRANS}\ \frac{\text{MUST-SI}\ \dfrac{A \boxdot\!\!\rightarrow c}{A \cup B \boxdot\!\!\rightarrow c}\ (var(c) - var(A)) \cap var(B) = \emptyset \qquad \text{MUST-SI}\ \dfrac{B \wedge c \boxdot\!\!\rightarrow \bot}{(A \cup B) \wedge c \boxdot\!\!\rightarrow \bot}}{A \cup B \boxdot\!\!\rightarrow \bot}$$

RESOLVE-$\bot$ was introduced in its quantifier free form in the section on KL_2. Although it deals with universally quantified constraint rules its derivation relies on EXISTS-$\bot$ and $\sim$-RIGHT from which it inherits restrictions on variables:

$$\text{EXISTS-}\bot\ \frac{\sim\text{-RIGHT}\ \dfrac{A \wedge c \boxdot\!\!\rightarrow \bot}{A \boxdot\!\!\rightarrow \overline{c}}\ var(c) \subseteq var(A) \qquad B \wedge \overline{c} \boxdot\!\!\rightarrow \bot}{A \cup B \boxdot\!\!\rightarrow \bot}$$

(and the symmetric form, which gives the variable restrictions quoted for RESOLVE-$\bot$ in Figure 6).

5.3 Semantics

A set of ground literals is consistent when it contains no complementary pair of literals $p(k_1, \ldots, k_n)$ and $\sim p(k_1, \ldots, k_n)$. Violation sets (sets of ground literals) are defined as in KL_2.

Given a (countable but not necessarily finite) set R of rules and a (finite) set A of ground literals, the consequences $out_3(R, A)$ are defined, as in KL_2, in terms of a set $aux_e^q(R)$ of additional rules representing the consequences of the inference rules for negation, $\sim$-LEFT and $\sim$-RIGHT. We will have:

$$out_3(R, A) = out_1^q(R \cup C_{\mathcal{P}} \cup aux_e^q(R),\ A)$$

$C_{\mathcal{P}}$ is the set of rules corresponding to $\sim$-LEFT:

$$\{p(x_1, \ldots, x_n) \wedge \sim p(x_1, \ldots, x_n) \boxdot\!\!\rightarrow \bot \mid p \in \mathcal{P},\ x_i \in \mathcal{X},\ \mathsf{arity}(p) = n\}$$

As usual we omit the subscript $\mathcal{P}$ when it is obvious from context.

$out_1^q(R, A)$ is the set of all possible outcomes obtained by applying the rules in R to the assumptions A. Each element of $out_1^q(R, A)$ is a set (finite if R

is finite) of ground literals. The definition is essentially the same as for KL_1 but adjusted to deal with variables in rules.

Notice that since R is a set of unground rules with variables and A is a set of grounded literals, it is no longer the case that assumptions A can be replaced by 'facts' (unconditional $\Box\!\!\rightarrow$ rules with singleton head). An expression $\top \Box\!\!\rightarrow a$ where a is a grounded literal is not a valid rule in KL_3 (unless a is a 0-ary term).

Definition 15 Let R be a set of KL_3 rules and A a set of ground literals.

$$out_1^q(R, A) = \{X \in cns^q(R, A) \mid X \models R\}$$

$$\begin{aligned} cns_0^q(R, A) &= \{A\} \\ cns_{n+1}^q(R, A) &= \{X \cup \{t\} \mid X \in cns_n^q(R, A),\ t \in step^q(R, X)\} \\ cns^q(R, A) &= \bigcup_{n \geq 0} cns_n^q(R, A) \end{aligned}$$

$$step^q(R, X) = \{c.\theta \mid B \Box\!\!\rightarrow C \in R \text{ or } B \Diamond\!\!\rightarrow C \in R,\ B.\theta \subseteq X,\ c \in C,\ c.\theta \text{ is ground}\}$$

The $step^q$ function takes a set of rules and a set of ground literals and produces all the ground literals that can be inferred in a single step using a single rule from R. $step^q$ is exactly like the $step$ function in the definition of cns and out_1 for KL_1, except for the need to instantiate variables in rules to constants in the ground literals of argument X. The substitution θ can include new fresh constants that do not appear in A that can serve as witnesses for existentially quantified variables.

Example 19 Suppose $R = \{p(x) \Box\!\!\rightarrow \exists y\, q(x, y)\}$ and $A = \{p(a)\}$.

$$step^q(R, A) = \{q(a, a), q(a, v_0), q(a, v_1), \dots\}$$

Here, v_0 and v_1 are new fresh constants. We assume we have an infinite stock of such constants $v_0, v_1, \dots$.

$aux_e^q(R)$ is defined analogously to $aux_e(R)$ in KL_2. For rules without existential heads this will be exactly as for KL_2 with universal rules treated as standing for the set of their ground instances. The additional ingredient for existential rules is an application of the inference rule EXISTS-$\perp$ as discussed informally in the previous section.

Definition 16 Let R be a set of KL_3 rules. $must_\perp(R)$, $resolve_\perp^*(R)$ and $right^+(R)$ are the three derived rules in Figure 6, defined as for KL_2 (and in accordance with the relevant KL_3 variable restrictions). Let $exists_\perp^*(R)$ denote the closure of rules R under EXISTS-$\perp$. Define:

$$aux_e^q(R) = right^+(resolve_\perp^*(exists_\perp^*(R \cup must_\perp(R))))$$

$R \cup aux_e^q(R)$ is the closure of R under RIGHT-$\sim$, EXISTS-$\bot$ and MUST-$\bot$. In $aux_e^q(R)$ it is sufficient to perform a single application of MUST-$\bot$, which deals with non-existential rules, and then the closure under EXISTS-$\bot$ and RESOLVE-$\bot$. The latter can be done in two separate steps, first the closure under EXISTS-$\bot$ and then the closure under RESOLVE-$\bot$. This is because (as was shown earlier) RESOLVE-$\bot$ is derivable as RIGHT-$\sim$ followed by EXISTS-$\bot$: $resolve_\bot(R) = exists_\bot(R \cup right(R))$ for any R. $resolve_\bot(R)$, for any R, is already closed under $exists_\bot$.

Example 20 Suppose:

$$R = \begin{cases} p(x) \boxminus\!\!\rightarrow q(x) \\ \top \boxminus\!\!\rightarrow \exists x \sim q(x) \end{cases}$$
$$A = \{\}$$

Then

$$aux_e^q(R) = \begin{cases} p(x) \boxminus\!\!\rightarrow q(x) \\ \sim q(x) \boxminus\!\!\rightarrow \sim p(x) \\ p(x) \wedge \sim q(x) \boxminus\!\!\rightarrow \bot \end{cases}$$

$$out_3(R, A) = \begin{cases} \{\sim p(v_0), \sim q(v_0)\}, \\ \{\sim p(v_0), \sim q(v_0), \sim p(v_1), \sim q(v_1)\}, \\ \{\sim p(v_0), \sim q(v_0), \sim p(v_1), \sim q(v_1), \sim p(v_2), \sim q(v_2)\}, \\ \ldots \end{cases}$$

Note that the existentially quantified variable x in the rule $\top \boxminus\!\!\rightarrow \exists x \sim q(x)$ of R appears in the body of the inferred constraint rule $p(x) \wedge \sim q(x) \boxminus\!\!\rightarrow \bot$ of $aux_e^q(R)$. The variable restrictions in EXISTS-$\bot$ however do *not* sanction the inference of the rule $p(x) \boxminus\!\!\rightarrow \bot$.

5.4 Equality

We add an extra binary logical operator $\neq$. The expression $x \neq y$ does not represent the act of subsuming x and y under the mark of inequality. Rather, $\neq$ is a testing operator that is different from the act of subsumption: to test if $x \neq y$ is just to see whether the denotations of x and y are distinct. Expressions of the form $x \neq y$ can appear only in the body of a rule; x and y must be variables appearing in the body.

One can think of a rule as having two distinct components (T, r) where r is an expression of the form $B \boxminus\!\!\rightarrow C$ or $B \diamond\!\!\rightarrow C$, and T is a set, possibly empty, of $\neq$ tests on variables appearing in B. However, for readability, we allow the inequality tests in T to be written in the body of a rule as if they were atoms.

Example 21 Suppose $R = \{p(x) \boxrightarrow \exists y\, q(x,y)\}$ and $A = \{p(a)\}$.

$$out_3(R,A) = \begin{cases} \{p(a), q(a,a)\} \\ \{p(a), q(a,\nu_0)\} \\ \{p(a), q(a,a), q(a,\nu_0)\} \\ \{p(a), q(a,\nu_0), q(a,\nu_1)\} \\ \dots \end{cases}$$

If we add an extra rule containing $\neq$, then we can constrain the set of witnesses.

$$R = \begin{cases} p(x) \boxrightarrow \exists y\, q(x,y) \\ q(x,y) \wedge q(x,z) \wedge y \neq z \boxrightarrow \bot \end{cases}$$
$$A = \{p(a)\}$$
$$out_3(R,A) = \begin{cases} \{p(a), q(a,a)\} \\ \{p(a), q(a,\nu_0)\} \\ \{p(a), q(a,\nu_1)\} \\ \{p(a), q(a,\nu_2)\} \\ \dots \end{cases}$$

Note that $\bot$-RIGHT does not allow us to infer from the rule $q(x,y) \wedge q(x,z) \wedge y \neq z \boxrightarrow \bot$ in R to the rule $q(x,z) \wedge y \neq z \boxrightarrow \sim q(x,y)$. In the (T,r) representation described above, that would be an inference from $(\{y \neq z\},\ q(x,y) \wedge q(x,z) \boxrightarrow \bot)$ to $(\{y \neq z\},\ q(x,y) \boxrightarrow \exists z\, q(x,z))$, which does not satisfy the variable restrictions of $\bot$-RIGHT.

To handle inequality, we modify what it means for a set X of ground literals to satisfy the body of a rule to take into account the possible presence of $\neq$ tests. Let us think of the inequality tests as belonging to the body, B. We will say that X satisfies B with ground instantiation of variables θ, written $X \models_\theta B$, if for every literal b in B, $b.\theta \in X$, and for every expression $x \neq y$ in B, the constants $x.\theta$ and $y.\theta$ are distinct. We are thereby making a unique names assumption on constants: two constants denote distinct objects when they are lexicographically distinct.

The adjustment for $step^q$ is as follows:

$$step^q(R,X) = \{c.\theta \mid B \boxrightarrow C \in R \text{ or } B \diamondrightarrow C \in R,\ X \models_\theta B,\ c \in C,\ c.\theta \text{ is ground}\}$$

Example 22 This example shows how the natural numbers can be constructed. The rules R are:

$$\begin{aligned}
\top \;&\Box\!\!\rightarrow\; \exists x\, zero(x)\\
zero(x) \;&\Box\!\!\rightarrow\; nat(x)\\
zero(x) \wedge zero(y) \wedge x \neq y \;&\Box\!\!\rightarrow\; \bot\\
nat(x) \;&\Diamond\!\!\rightarrow\; \exists y\, succ(x,y)\\
succ(x,y) \wedge succ(x,z) \wedge y \neq z \;&\Box\!\!\rightarrow\; \bot\\
succ(x,y) \;&\Box\!\!\rightarrow\; nat(y)\\
succ(x,x) \;&\Box\!\!\rightarrow\; \bot\\
succ(x,y) \;&\Box\!\!\rightarrow\; less(x,y)\\
succ(x,y) \wedge less(y,z) \;&\Box\!\!\rightarrow\; less(x,z)\\
less(x,x) \;&\Box\!\!\rightarrow\; \bot
\end{aligned}$$

Note the $\Diamond\!\!\rightarrow$ rule for constructing successors. These rules allow us to create any finite subset of the natural numbers.

For example, one of the members of $out_3(R, \emptyset)$ is:

$$\begin{array}{rl}
nat(v_0) & \\
nat(v_1) & \\
nat(v_2) & \\
zero(v_0) & \sim zero(v_1)\\
succ(v_0, v_1) & \sim zero(v_2)\\
succ(v_1, v_2) & \sim succ(v_1, v_1)\\
less(zero, v_1) & \sim succ(v_1, v_0)\\
less(zero, v_2) & \sim succ(v_0, v_2)\\
less(v_1, v_2) & \sim succ(v_0, v_0)
\end{array}$$

5.5 Comparing KL_3 with geometric logic

All rules in KL_3 are of the form $\forall \bar{x}\, \phi(\bar{x}) \circ\!\!\rightarrow \exists \bar{y}\, \psi(\bar{x}, \bar{y})$, where $\bar{x}$ and $\bar{y}$ are tuples of variables and $\circ\!\!\rightarrow$ is either $\Diamond\!\!\rightarrow$ or $\Box\!\!\rightarrow$. These rules have the same quantifier structure as the rules of geometric logic[36].

The geometric formulae (also known as the "coherent implications") are the implications $C \rightarrow D$ where

$$\begin{aligned}
C &::= \top \mid C \wedge P\\
D &::= \bot \mid D \vee E\\
E &::= \exists \bar{x}\, C
\end{aligned}$$

[36] The importance of geometric logic for understanding Kant's thought is stressed in the pioneering papers by Theodora Achourioti and Michiel van Lambalgen [1,2]. For geometric logic in general, see [14], [24], [5], [6], [44].

and P ranges over $\mathcal{L}$[37].

Although the rules of KL_3 have the same quantifier structure as the rules of geometric logic, there are a number of differences. First, KL_3 has two types of rule, $\Box\!\!\rightarrow$ and $\Diamond\!\!\rightarrow$, while geometric logic has only one. Second, KL_3 has predicate negation, while geometric logic does not include any sort of negation. Third, weakening the output is valid in geometric logic[38], but not in KL_3.

The fourth difference between the two systems is the way in which the tree of nodes[39] is generated. In KL_3, to generate the successors $step^q(R, X)$ of a set X of atoms, we consider all rules in R whose bodies are satisfied. When we have finished constructing the nodes, we filter them to accept only those that satisfy all the $\Box\!\!\rightarrow$ rules. In geometric logic, the dynamical proof tree is generated by considering only violated rules: rules whose body is satisfied but whose head is unsatisfied. To see the difference, consider the rule-set R consisting of only one rule:

$$\top \Box\!\!\rightarrow \exists x\, \phi(x)$$

In geometric logic, the proof tree contains one node with the single atom $\phi(a_0)$ for some constant a_0. Once the rule's head has been satisfied, it is no longer available to generate further nodes. In KL_3 by contrast, the $step^q$ function allows a rule to be applied whenever its body is satisfied, so $out_3(R, \{\})$ contains infinitely many possible solutions: $\{\phi(a_0)\}, \{\phi(a_0), \phi(a_1)\}, \{\phi(a_0), \phi(a_1), \phi(a_2)\}, \ldots$.

A final difference is that KL_3 is a logic of conditional imperatives relating actions that do not have truth values, while geometric logic is a truth-functional logic.

All of these differences are crucial to the intended application of our logic in understanding Kant (see Sections 2, 3.7, and 6).

5.6 Translating natural language into KL_3

Finally in this section, and before returning to Kant's texts, we shall spend a little time showing how natural language sentences can be translated into KL_3. This exercise is important because the translation guidelines for KL_3 are rather different from those for translating natural language into first-order logic.

5.6.1 Singular judgements

In first-order logic, a singular judgement, such as "Caius is mortal," is translated into an atom:

$$mortal(caius)$$

[37] Note that geometric logic, unlike KL_3, does allow constants as terms in rules.

[38] See the classical evaluation rule for disjunction on page 3 of [14]: $X \Vdash \phi_1 \vee \phi_2$ if $X \triangleleft U$ and for all $Y \in U$, $Y \Vdash \phi_1$ or $Y \Vdash \phi_2$.

[39] Each "node" is a set of literals in $out_n^q(R, A)$ at depth n.

where *mortal* is a one-place predicate and *caius* is a constant representing the individual Caius.

In KL_3, by contrast, an atom represents a *subsumption* – the act of subsuming a private mental intuition under a mark. So in KL_3, declarative sentences are never translated into atoms (subsumptions). Instead, the judgement "Caius is mortal" is rendered as a *rule*:

$$caius(x) \boxdot\!\!\rightarrow mortal(x)$$

This is a conditional imperative that relates *actions*. It says: for all intuitions x, if you are subsuming x under the mark "caius", then also subsume x under the mark "mortal"!

Now a proper noun, such as "Caius," is normally taken to imply existence (there is at least one individual denoted by "Caius") and uniqueness (there is at most one individual denoted by "Caius"). If we wish to express existence and uniqueness in KL_3, we write:

$$\top \boxdot\!\!\rightarrow \exists x\, caius(x)$$
$$caius(x) \wedge caius(y) \wedge x \neq y \boxdot\!\!\rightarrow \bot$$

Judgements involving binary predicates are represented similarly. "Jack loves Jill" is rendered as:

$$jack(x) \wedge jill(y) \boxdot\!\!\rightarrow loves(x, y)$$

plus existence and uniqueness constraints, as needed.

5.6.2 All and some

Universally quantified judgements, such as "All men are mortal," are rendered directly into KL_3 as:

$$man(x) \boxdot\!\!\rightarrow mortal(x)$$

Recall, once more, that this rule is a conditional imperative stating what actions you must do: if you are subsuming private mental intuition x under the mark "man" then also subsume x under "mortal"!

Judgements involving "some" can be translated into KL_3 in two different ways. "Some men are fickle," for example, can be translated into:

$$man(x) \diamond\!\!\rightarrow fickle(x)$$

This is a permissive rule: if you are subsuming intuition x under "man", then feel free to also subsume x under "fickle"! This way of translating the sentence has no existential import whatsoever. It is fully compatible with there actually being no men at all. The other way of translating "Some men are fickle," by contrast, provides existential import:

$$\top \boxdot\!\!\rightarrow \exists x\, p_1(x)$$
$$p_1(x) \boxdot\!\!\rightarrow man(x)$$
$$p_1(x) \boxdot\!\!\rightarrow fickle(x)$$

Here, p_1 is a new predicate mark introduced to represent the conjunction of *man* and *fickle*, since conjunctions cannot be expressed directly in the conclusions of rules. These rules mean: you must construct at least one intuition x and subsume x under both "man" and under "fickle."

In [*Jäsche Logic* §46], Kant says that universal judgements ("all" judgements) imply particular judgements ("some" judgements). He is clearly thinking here of the inference from:

$$man(x) \boxdot\!\!\rightarrow mortal(x)$$

to:

$$man(x) \diamond\!\!\rightarrow mortal(x)$$

which is valid in KL_3 (using the MUST-MAY inference rule). He is not thinking of the inference from

$$man(x) \boxdot\!\!\rightarrow mortal(x)$$

to[40] :

$$\top \boxdot\!\!\rightarrow \exists x\, man(x) \wedge mortal(x)$$

which is invalid (see Section 6).

One of the key strengths of first-order logic is its ability to handle multiply quantified sentences. We can infer, for example, from "there is some (particular) prince who has offended every delegate", that "for every delegate, there is some prince who has offended her". Aristotle's two-term logic has been rightly criticised for its inability to deal with inferences involving multiply quantified sentences. KL_3 does not suffer from the inadequacies of Aristotle's logic. A single rule in KL_3 is implicitly of the form:

$$\forall \bar{x}\, \phi(\bar{x}) \circ\!\!\rightarrow \exists \bar{y}\, \psi(\bar{x}, \bar{y})$$

where $\bar{x}$ and $\bar{y}$ are tuples of variables, and $\circ\!\!\rightarrow$ is either $\diamond\!\!\rightarrow$ or $\boxdot\!\!\rightarrow$. A single rule cannot capture a sentence of the form $\exists \bar{x} \forall \bar{y}\, \phi(\bar{x}, \bar{y})$. However, a *set* of rules in KL_3 can capture this. For example, "there is some (particular) prince who has offended every delegate" can be rendered as R_1 below, while "for every

[40] Strictly speaking, the consequent is ill-formed as conjunctions are not permitted in the conclusions of rules. But rules with conjunctive conclusions can always be translated by introducing an auxiliary predicate, for example:

$$\begin{aligned} \top &\boxdot\!\!\rightarrow \exists x\, p(x) \\ p(x) &\boxdot\!\!\rightarrow man(x) \\ p(x) &\boxdot\!\!\rightarrow mortal(x) \end{aligned}$$

delegate, there is some prince who has offended her" can be rendered as R_2:

$$R_1 = \begin{cases} \top \boxdot\!\!\rightarrow \exists x\, p_1(x) \\ p_1(x) \boxdot\!\!\rightarrow \mathit{prince}(x) \\ p_1(x) \wedge \mathit{delegate}(y) \boxdot\!\!\rightarrow \mathit{offended}(x, y) \end{cases}$$

$$R_2 = \begin{cases} \mathit{delegate}(y) \boxdot\!\!\rightarrow \exists x\, p_2(x, y) \\ p_2(x, y) \boxdot\!\!\rightarrow \mathit{prince}(x) \\ p_2(x, y) \boxdot\!\!\rightarrow \mathit{offended}(x, y) \end{cases}$$

In the above example R_2 is a **conservative extension** of R_1 in the following sense:

1. for all A and X, if $X \in \mathit{out}_3(R_1, A)$ then $\exists Y \in \mathit{out}_3(R_1 \cup R_2, A)$ such that $X \cap Y = X$;
2. for all A and Y, if $Y \in \mathit{out}_3(R_1 \cup R_2, A)$ then $\exists X \in \mathit{out}_3(R_1, A)$ such that $X \cap Y = X$.

More generally, [15] shows that, for each set F of first-order sentences, there is a set of sentences of geometric logic that is a conservative extension of F[41].

5.6.3 The "is" of identity and the "is" of predication

In first-order logic, the sentence "Phosphorus is bright" is translated as a predication:

$$\mathit{bright}(\mathit{phosphorus})$$

where *bright* is a one-place predicate and *phosphorus* is a constant. The sentence "Hesperus is Phosphorus," by contrast, involves the "is" of identity and should be translated as:

$$\mathit{hesperus} = \mathit{phosphorus}$$

If we wish to infer that, therefore, Hesperus is bright, we need to use Leibniz's law. This is an (infinite) axiom schema licensing, for every sentence $\phi(x)$ with one free variable x the inference:

$$\textsc{Leibniz Law}\ \frac{\phi(x) \quad x = y}{\phi(y)}$$

In KL_3, by contrast, the two senses of "is" do not come apart. "Phosphorus is bright" is translated as:

$$\mathit{phosphorus}(x) \boxdot\!\!\rightarrow \mathit{bright}(x)$$

[41] Many commentators (for example, MacFarlane [40], p.26; also [19] and [53]) assume or claim that Kant's logic does not support nested quantifiers, while our formalization presupposes that his logic does have this expressive power. Our main evidence that this common view is wrong is the systematic support our account gets from making sense of Kant's otherwise notoriously obscure and problematic Table of Judgements (section 6). But for compelling textual evidence, see [1], pages 260-2.

"Hesperus is Phosphorus" is rendered as:

$$hesperus(x) \boxdot\!\!\rightarrow phosphorus(x)$$

together with the symmetric rule:

$$phosphorus(x) \boxdot\!\!\rightarrow hesperus(x)$$

The inference to $hesperus(x) \boxdot\!\!\rightarrow bright(x)$ does not require any infinite axiom schema. It just involves the standard MUST-TRANS inference rule.

5.6.4 Two types of negation

Natural language distinguishes between sentence-negation (e.g. "It is not the case that Jack is tall") and predicate-negation[42] (e.g. "Jack is not tall"). First-order logic, of course, cannot capture these two distinct interpretations. The only negation in first-order logic is sentential negation. But KL_3 can capture the two distinct readings. "It is not the case that Jack is tall" is rendered as:

$$jack(x) \wedge tall(x) \boxdot\!\!\rightarrow \bot$$

"Jack is not tall" is rendered as:

$$jack(x) \boxdot\!\!\rightarrow {\sim}tall(x)$$

Now in KL_3 these two particular claims are provably equivalent – but in general, when existentially quantified variables are involved, sentence-negation and predicate-negation are *not* equivalent in KL_3. Consider "Jack is not married to anyone":

$$jack(x) \wedge married(x, y) \boxdot\!\!\rightarrow \bot$$

Compare with "There is someone who Jack is not married to":

$$jack(x) \boxdot\!\!\rightarrow \exists y{\sim}married(x, y)$$

Neither claim entails the other.

6 Recovering the Table of Judgements

The Table of Judgements [A70/B95] is divided into four "titles": Quantity, Quality, Relation, and Modality. Kant clearly thought this division was fundamental because it appears as an organising framework throughout the critical works[43]. Each title represents one structural feature of a judgement.

[42] As does Kant (in *transcendental* logic), e.g. at [A71-3/B97-8], [*Jäsche Logic* p. 103-4]. See also [56] p.268.

[43] See, for example, the Table of Categories [A80/B106], the four aspects of time determination [A145/B184], the four principles [A161/B200], the four ways of comparing concepts [A263/B319], the four aspects of the concept of nothing [A291-2/B348]. For Kant on the importance of his table, see [A80-1/B107-7], *Prolegomena* 4: 306. Kant also employs the four titles as an organising framework in the *Critique of Practical Reason* and in the *Critique of the Power of Judgement*. For a comprehensive visual representation of the extent of this organizing structure in the first *Critique*, see [REFERENCE REMOVED FOR ANONYMITY].

In Kant's table, there are three possible values for each structural feature, so there are at most 3^4 possible types of judgement[44]. The four titles were in widespread use in the logic textbooks of the time[45], but Kant's particular use of them was unusual.

The **Quantity** of a categorical subject-predicate judgement indicates whether the extension of the subject is partly or wholly contained in the extension of the predicate. If the extension of the subject S is wholly contained in the extension of the predicate P, then we say "all S are P", and the judgement has **universal** quantity. If the extension of S is only partly contained in the extension of P, then we say "some S are P", and the judgement has **particular** quantity. If the extension of S is a singleton, and this single element is a member of the extension of P, then we say "the individual S is P", and the judgement has **singular** quantity.

One problem with this way of characterising Quantity is that it only applies to categorical judgements involving monadic predicates. But we shall see, below, how to extend this idea naturally to all other types of judgement, including hypothetical and disjunctive judgements involving binary or n-ary predicates.

Kant claimed that singular judgements are a sub-type of universal judgements [A71/B96] [*Jäsche Logic* 9:102]. This was a common claim for logicians working within the Aristotelian two-term logic. But note that this claim is obviously false if universal and singular judgements are interpreted in terms of first-order logic. The singular judgement $p(a)$ is not a sub-type of universal judgement $(\forall x)\ a(x) \supset p(x)$.

The **Quality** of a judgement indicates whether the predicate is affirmed or denied of the subject. If the predicate is affirmed of the subject, as in "All men are mortal", then the judgement is **affirmative**. If the predicate is denied of the subject, as in "It is not the case that the soul is mortal", then the judgement is **negative**. But if the negation of the predicate is affirmed of the subject, as in "The soul is non-mortal", then the judgement is **infinite**[46]. The infinite judgements are, according to Kant, a sub-type of the affirmative judgements:

> If I had said of the soul that it is not mortal, then I would at least have avoided an error by means of a negative judgement. Now by means of the proposition "The soul is non-mortal" I have certainly made an actual affirmation as far as logical form is concerned, for I have placed the soul within the unlimited domain of undying things. [A72/B97]

Note that both the distinction between negative and infinite judgements, and the claim that the infinite judgements are a sub-type of affirmative judgements, make no sense within first-order logic. In Frege's logic and its descendants, there is only one type of negation: sentence-level negation.

[44] In practice, there will be slightly fewer, since some combinations are incompatible. For example, a judgement cannot both be negative and disjunctive. Nor can it be both negative and particular. See [2].

[45] See in particular *The Port-Royal Logic [4]*.

[46] "In negative judgements the negation always affects the copula; in infinite ones it is not the copula but rather the predicate that is affected" [*Jäsche Logic* 9:104]

Kant's use of **Relation** is very different from its current meaning. In modern logic, a relation is a n-ary predicate where $n > 1$. For Kant, the Relation is a structural feature of a judgement indicating how the various subsumptions in the judgement are related to each other:

> All relations of thinking in judgement are either those a) of the predicate to the subject, b) of the ground to the consequence, and c) between the cognition that is to be divided and all of the members of the division. [A73/B98]

In case (a), when a judgement involves just two subsumptions (e.g. "all men are mortal"), then the judgement is **categorical**. In case (b), when a judgement has a condition that must be satisfied (e.g. "If there is perfect justice, then obstinate evil will be punished"), then the judgement is **hypothetical**. In case (c), when a judgement has a disjunctive conclusion (e.g. "The world exists either through blind chance, or through inner necessity, through an external cause"), then the judgement is **disjunctive**[47] [A73-4/B98-9].

Strawson [53] criticised Kant's use of Relation for being neither exhaustive nor exclusive. The three types of Relation are not exhaustive since some types of judgement (e.g. conjunctions) are not present at all. The three types of Relation are not exclusive since hypotheticals and disjunctions can, in standard propositional logic, be inter-defined using negation: $p \supset q$ if and only if $\neg p \vee q$. However we shall see, below, that in KL_3, Kant's threefold division is very natural.

The fourth title, **Modality**, is a different type of feature than the others. While Quantity, Quality, and Relation are structural features of an individual judgement, Modality (as we read Kant) is a feature indicating how the judgement relates to the rest of the judgements held by an agent:

> The modality of judgements is a quite special function of them, which is distinctive in that it contributes nothing to the content of the judgement (for besides quantity, quality, and relation there is nothing more that constitutes the content of the judgement), but rather concerns only the value of the copula *in relation to thinking in general*. [A74/B100]

The Modality of a judgement can be either **problematic**, **assertoric**, or **apodictic**. These are not the alethic modalities of possibility, actuality, and necessity. They are more like epistemic modals that relate us to the alethic modalities in particular ways:

> Problematic judgments are those in which one regards the assertion or denial as merely possible (arbitrary). Assertoric judgments are those in which it is considered actual (true). Apodictic judgments are those in which it is seen as necessary. [A74-5/B100]

In the [*Jäsche Logic* 9:108-9], Kant goes on to explain his modalities of judgement in terms of the very same normative notions (may/must) that have been

[47] Disjunctions for Kant are *exclusive* disjunctions (see Section 3.1).

so central to KL_3. The difference, we shall see, is that they function at a different level.

In each of the four titles, the third moment is defined as a sub-type of the first moment. A singular judgement is a sub-type of universal judgement; an infinite judgement is a sub-type of affirmative judgement; a disjunctive judgement is a sub-type of categorical judgement, and an apodictic judgement is a sub-type of problematic judgement. According to Kant, the third moment in each title entails a judgement of the second moment. A singular judgement entails a particular judgement; an infinite judgement entails a negative judgement; a disjunctive judgement entails a hypothetical judgement, and an apodictic judgement entails an assertoric judgement.

Kant's Table of Judgements has been roundly criticised for being incomplete, confused, or for being based on an impoverished expressively-limited logic. In this paper we argue, by contrast, that KL_3 is a powerful and expressive logic in which Kant's table emerges as the most natural way of categorising rules.

6.1 KL_3 Makes sense of Kant's Table of Judgements

Since Kant sees a judgement as a type of rule (see Sections 1 and 2), a way of classifying rules will also be a way of classifying judgements. In this section, we shall provide four ways of classifying rules in KL_3, and show how each classification corresponds to one of the four titles in the Table of Judgements.

Quantity. In KL_3, there are two types of rule : conditional imperatives and conditional permissives. An imperative of the form $p \boxdot\!\!\rightarrow q$ means: "if you are performing p, then also perform q!" A permissive of the form $p \Diamond\!\!\rightarrow q$ means: "if you are performing p, then feel free to also perform q!"

We propose the following simple identification: a rule (judgement) has universal quantity if it is a conditional imperative, while a rule has particular quantity if it is a conditional permissive. So, for example, the universal judgement "all men are mortal" would be rendered as:

$$man(x) \boxdot\!\!\rightarrow mortal(x)$$

while the particular judgement "some men are fickle" would be rendered as:

$$man(x) \Diamond\!\!\rightarrow fickle(x)$$

A singular judgement is a sub-type of universal judgement in which there is at most one object falling under the subject term. "Caius is mortal", for example, would be rendered by a *pair* of rules:

$$caius(x) \boxdot\!\!\rightarrow mortal(x)$$
$$caius(x) \wedge caius(y) \wedge x \neq y \boxdot\!\!\rightarrow \bot$$

This way of characterising Quantity has three appealing features. First, it shows how a singular judgement can be a type of universal judgement. In first-order logic, by contrast, a singular judgement is typically *not* rendered as a type of universal judgement. Second, it shows how Quantity can apply to *all* types of judgement. Recall that Quantity is normally defined for affirmative categorical judgements involving monadic predicates (subject-predicate sentences of the form "S is P"), and there is a problem how to extend this definition to *all* types of judgement. If Quantity is based on the distinction between conditional imperatives and conditional permissives, then it applies to *all* types of rule. Finally, this way of defining Quantity shows why Kant thought[48] that the inference from universal to particular quantity is valid. Consider the inference:

All S are P
Therefore, some S are P

If these statements are translated into first-order logic, the inference is obviously invalid: we cannot infer from $\forall x\ s(x) \supset p(x)$ that $\exists x\ s(x) \wedge p(x)$ since there may be no objects whatsoever satisfying $s(x)$. However, when we translate into KL_3, we get:

$$s(x) \boxdot\!\!\rightarrow p(x)$$
$$s(x) \Diamond\!\!\rightarrow p(x)$$

We can infer $s(x) \Diamond\!\!\rightarrow p(x)$ from $s(x) \boxdot\!\!\rightarrow p(y)$ using the MUST-MAY inference rule.

In Kant's Table of Judgements, the third moment always entails a judgement of the second moment. In the case of Quantity, a singular judgement is a type of universal judgement, which itself entails a particular judgement, using the MUST-MAY inference rule.

Quality. In KL_3, a conditional imperative has the form $p_1 \wedge ... \wedge p_n \boxdot\!\!\rightarrow q_1 \vee ... \vee q_m$. In particular, if the disjunction is empty, then the imperative acts as a *constraint*: $p_1 \wedge ... \wedge p_n \boxdot\!\!\rightarrow \bot$, in other words: whatever you do, do not perform all of $p_1, ..., p_n$. Constraints can be used to represent negative judgements[49]:

$$jack(x) \wedge married(x) \boxdot\!\!\rightarrow \bot$$

represents the judgement that it is not the case that Jack is married. Recall from Section 5 that the set of predicates in KL_3 contains positive and negative marks. Given a set $\mathcal{P}$ of predicate marks, the complete set $\mathcal{P}^{+/-}$ of signed predicates is:

$$\mathcal{P}^{+/-} = \mathcal{P} \cup \{\sim p \mid p \in \mathcal{P}\}$$

An infinite judgement is an affirmative judgement in which the conclusion involves a negated mark. To say that Jack is un-married, we write:

$$jack(x) \boxdot\!\!\rightarrow \sim married(x)$$

[48] See e.g. [*Jäsche Logic* 9:116].

[49] "Negative judgements have the special job of preventing error" [A709/B737].

Note that the negation binds to the mark *married* and not to the subsumption *married*(x).

Unlike first-order logic, KL_3 is able to distinguish between negative and infinite judgements, and is able to characterise infinite judgements as a subtype of affirmative judgements.

In Kant's Table of Judgements, the third moment always entails a judgement of the second moment. In the case of Quality, an infinite judgement (e.g. $p \boxrightarrow \sim q$) entails a negative judgement (e.g. $p \wedge q \boxrightarrow \bot$) using the following inference:

$$\dfrac{\dfrac{p \boxrightarrow \sim q}{p \wedge q \boxrightarrow \sim q}\,\text{SI} \qquad \dfrac{\dfrac{}{q \wedge \sim q \boxrightarrow \bot}\,\sim\text{-LEFT}}{p \wedge q \wedge \sim q \boxrightarrow \bot}\,\text{SI}}{p \wedge q \boxrightarrow \bot}\,\text{MUST-TRANS}$$

Relation. In KL_3, imperatives of the form $p_1 \wedge ... \wedge p_n \boxrightarrow q_1 \vee ... \vee q_m$ and permissives of the form $p_1 \wedge ... \wedge p_n \diamond\!\!\rightarrow q_1 \vee ... \vee q_m$ can be categorised based on how many conjuncts n they have in the antecedent, and how many disjuncts m they have in the consequent. If there is one element in the antecedent, then the rule is categorical. If there are many elements in the antecedent, then the rule is hypothetical[50]. If there is one element in the antecedent, but many elements in the consequent, then the rule is disjunctive [A93-4, B98-9].

Note that Strawson's criticism of Kant's three moments of Relation (that they are not exhaustive) does not apply to this formalization in KL_3. The first two types of rule are exhaustive as long as $n > 0$.

Recall that in Kant's Table of Judgements, the third moment of each title is a sub-type of the first moment, and entails a judgement of the second moment. In the case of Relation, the disjunctive judgement (because it has one element in the antecedent) is a sub-type of the categorical and entails a hypothetical judgement (using MUST-SI or MAY-SI).

Modality. The Kantian agent makes sense of its sensory perturbations by constructing and applying rules. These rules are conditional imperatives and permissives relating mental acts, e.g., for all private mental intuitions x, if you are subsuming x under mark p, then also subsume x under mark q!

[50] Compare [*Kant and the Capacity to Judge*, p. 103n] where Longuenesse characterises categoricals as implications of the form $\forall x\ b(x) \supset d(x)$ and hypotheticals as implications of the form $\forall x\ b(x) \wedge c(x) \supset d(x)$. There are three important differences between her approach and ours. First, Longuenesse (p.93n) is forced to retreat to the claim, unsupported by the text, that only universal judgements are rules, whereas we can make sense of Kant's claim that all judgements are rules. Second, with support from the text, we make use of conditional permissives, which Longuenesse does not consider. Third, and most importantly, her formalization uses first-order logic, relating propositions that have truth-values, while the rules in KL_3 are *conditional imperatives relating actions that do not have truth-values*. In [A73-4, B98-9] and [B141], Kant uses a slightly different form for hypothetical judgements, in which a hypothetical judgement contains other judgements as *constituents*. This is different from how hypotheticals are treated in KL_3 and in [*Kant and the Capacity to Judge*, p. 103n], where they are treated as rules with multiple elements in the antecedent or consequent. We have explored an extension of KL_3 that includes embedded rules, and will describe this fully in future work.

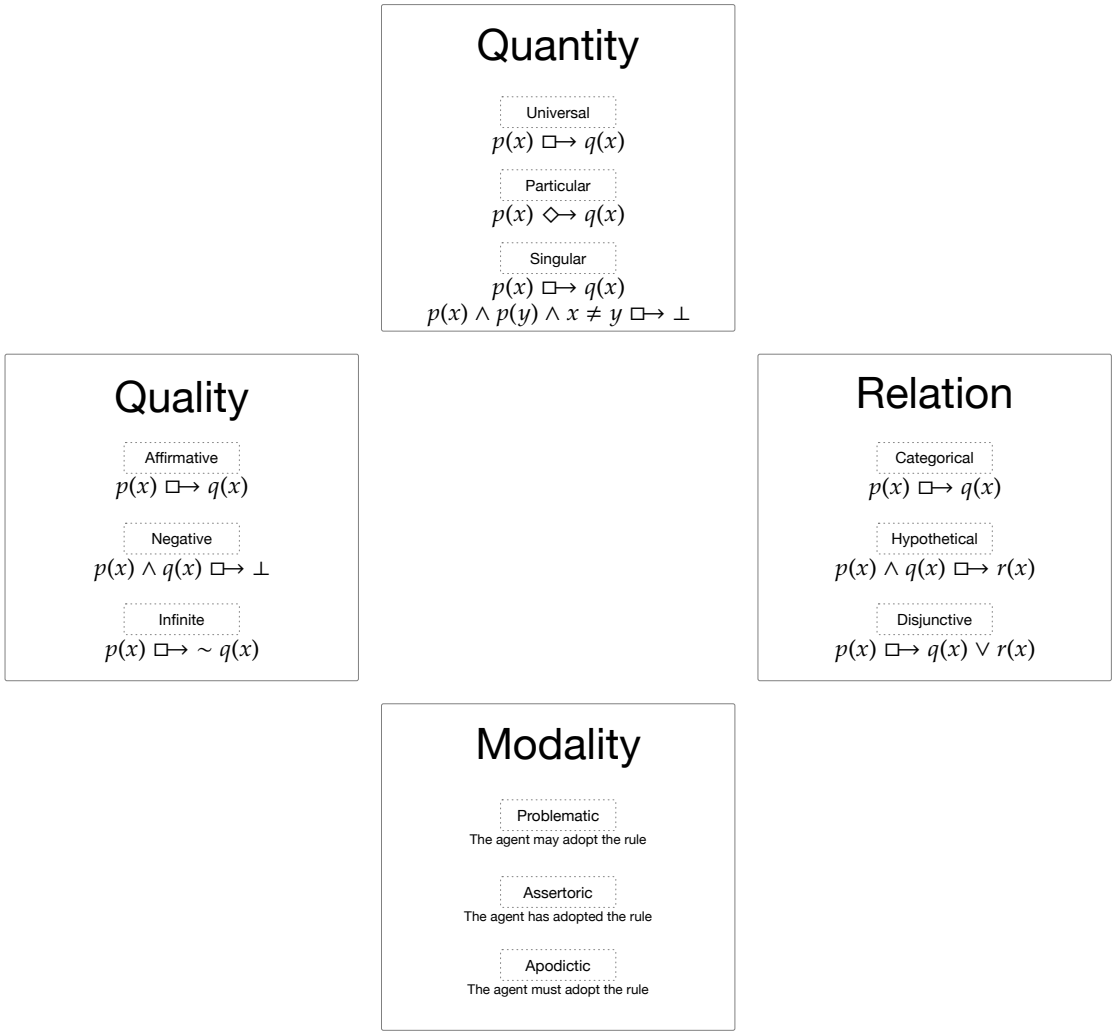

Fig. 7: Interpreting the Table of Judgements in KL

At any moment, the Kantian agent has a set A of subsumptions that it is performing, and a set R of rules that it has adopted. Given the subsumptions A and rules R, there are various different bundles of mental activity that are compatible with A and R. These are the various sets $X \in out(R, A)$, the various sets of subsumptions it may perform. There is also the one distinguished set of subsumptions it is actually performing. If we take the intersection of all the X such that $X \in out(R, A)$, then we get the subsumptions it must perform.

As well as the collections of subsumption acts that it may or must perform, however, there are also the *rules* that it may or must adopt. These are the selfsame normative notions at work in both cases – they have their force and

content relative to the agent's goal of achieving experience. But they function at different levels. Whereas the normative characterizations of subsumptions within rules were the basis of the types of Quantity, Quality, and Relation in the Table of Judgements, it is the normative characterizations of rules themselves that are the basis of the types of Modality.

Given a set A of subsumptions it is performing, and a set R of rules it has adopted, there are various further rules that it *may* adopt. Of course, not every set of rules can be added. Some rules may be incompatible with one of the existing rules in R. Or some rule may be incompatible with some of its current subsumptions. For example, if it is subsuming an intuition k under mark p, and also is subsuming k under q, then it may not adopt the rule:

$$p(x) \wedge q(x) \boxdot\!\!\rightarrow \bot$$

We propose that the problematic judgements are the rules an agent *may* adopt[51]:

> Problematic judgements are those in which one regards the assertion or denial as merely possible. [A74/B100]

Given a set R of rules that an agent has already adopted, the assertoric judgements are the rules in R that it has *already* committed to:

> The assertoric proposition ... indicates that the proposition is already bound to the understanding according to its laws. [A76/B101]

Further, the apodictic judgements are the rules that it *must* adopt[52], given R:

> Apodictic judgements are those in which it [the judgement] is seen as necessary. [A75/B100]

These are the rules in $deriv(R)$, in Section 3.5 above. A summary of our interpretation of Kant's Table of Judgements in KL is given in Figure 7.

There are, then, two different levels in Kant's theory at which the same normative notions play a key role: there are the subsumptions that may / must be performed, and there are the judgements (i.e. rules) that may / must be adopted. These two levels can come apart: an agent *may* choose to adopt a rule saying that it *must* perform a particular subsumption. In this case, there is a sense in which the subsumption is necessary, even though the rule that prompted it need not have been adopted and is, hence, contingent:

> this word [the copula "is"] designates the relation of the representations to the original apperception and its **necessary unity**, even if the judgement itself is empirical, hence contingent [B142]

This passage brings to the fore a topic that has been latent in much of the preceding, but which we can only discuss very briefly here insofar as it relates to an important simplification in the present account of Kant's modalities of

[51] See also [*Jäsche Logic* 9:109]: "The soul of man may be immortal."
[52] See also [*Jäsche Logic* 9:109]: "The soul of man must be immortal."

judgement. The topic is unity of apperception. The simplification is that we have so far avoided the question of whether, and how, the Kantian agent can reject or revise rules it has previously adopted.

For Kant, unity of apperception and experience are two sides of the same coin. On our interpretation, the Kantian agent "binds" itself in two distinct but related senses when it constructs and applies its rules in constructing experience. First, it binds itself to its rules: it commits to up-holding those rules. Second, it binds itself together: it forms itself into a unity by up-holding its rules[53]. Now, the agent can only do either of these things insofar as it also binds its subsumptions into a unity, since the rules to and by which it binds itself just are procedures for generating subsumptions from subsumptions. And as we've said, if various (meta-) constraints on this activity are satisfied, this rule-bound unity of subsumptions will constitute experience. It is in this way that a unity of consciousness arises alongside and necessarily accompanies that consciousness of unity that is our experience of a coherent, unified external world. Unity of apperception and experience are two sides of the same coin. Both are the upshot of self-legislation.

So how does this relate to rule-revision? Above we assumed that the set R of rules an agent has adopted is fixed. Our prototype computer simulations [17, 18] of the Kantian cognitive architecture make the same assumption. But what if we consider the set of rules to be changing – what if we consider adding to or removing from R? Clearly some sort of revision will sometimes be required in the light of new information. Indeed, the Kantian agent will always be changing its rules to best account for the stream of sensory data. The pattern is new in every moment, constantly requiring revision in making coherent, unified sense of the on-going stream of sensory perturbations. However, if the Kantian agent can reject or revise any rule whenever it sees fit, then this makes a nonsense of the idea that the agent has previously *committed* to that rule (and with it, the quoted notion of necessary unity of apperception even in contingency of judgement). In what sense is the agent really bound to its rules or together into a unity? Kant's notion of spontaneity cannot just be a free-for-all – rather, it must be compatible with self-legislation properly so-called.

A thorough model of rule revision must answer two questions. First, under what circumstances is the agent permitted to revise a rule? Second, when it is in one of these special circumstances in which it is permitted to revise a rule, what is the proper procedure for revision? What are the constraints on acceptable revision? We do not here have space for a full answer or its formal implementation – that is a task for future work. But in brief: first, the agent is permitted to revise a rule when its current rules cannot make sense of its current sensory stimulations; second, the only acceptable revisions of a ruleset are revisions in which all previous subsumptions are still licensed. The end towards which the agent's activity is directed is experience (and

[53] Note that there is also a sense in which the agent binds others (and itself to others) in this way, in that its rules quantify over all intuitions (see Sections 2.2 and 5).

apperception), a coherent, unified representation. It cannot revise a rule-set if the new rule-set no longer legitimizes one of the activities it has already performed. (See Section 3.11.)

7 Conclusion

The Kantian agent is a self-legislating rule-induction system. It makes sense of its sensory perturbations by spontaneously constructing and applying rules. If this activity satisfies various constraints, the agent achieves experience: it has constructed a coherent, unified representation of a coherent, unified external world. We have defined a logic of conditional imperatives and permissives that was designed as a formalization of Kant's conception of rules relating mental acts that do not have truth-values. This logic includes but is not exhausted by an account of entailment relations between elements with truth-values. At its heart are the normative notions captured by conditional imperatives and permissives, rather than the notion of truth.

In this paper, we showed how the rules formalized in our logic have structural features that correspond precisely to those displayed in Kant's Table of Judgements. We also explained how this logic handles the major deontic paradoxes, how it differs from related logics, and how it translates natural language sentences. Of course our claim has not been that Kant had this precise logic in mind, but rather that it is based on, compatible with, and helps to explain part of Kant's view in the *Critique of Pure Reason* (and associated texts).

References

1. Achourioti, Theodora, and Michiel van Lambalgen. (2011). A formalization of Kant's transcendental logic. *The Review of Symbolic Logic*, 4.02 : 254-289.
2. Achourioti, Theodora, and Michiel van Lambalgen. (2012). Kant's logic revisited. *PhML-2012*.
3. Allais, Lucy. (2016). Conceptualism and Nonconceptualism in Kant: A Survey of the Recent Debate. In *Kantian Nonconceptualism* (pp. 1-25). Palgrave Macmillan, London.
4. Arnauld, Antoine, and Pierre Nicole. (1996). *Logic or the art of thinking*. Cambridge University Press.
5. Bezem, Marc, and Thierry Coquand. (2005). Automating coherent logic. *International Conference on Logic for Programming Artificial Intelligence and Reasoning*. Springer, Berlin, Heidelberg.
6. Bezem, Marc. (2005). On the undecidability of coherent logic. *Lecture notes in computer science*, 3838: 6.
7. Brandom, Robert B. (2008). *Between saying and doing: towards an analytic pragmatism*. Oxford University Press.
8. Brandom, Robert B. (2015). *From Empiricism to Expressivism*. Harvard University Press.
9. Brandom, Robert B. (2009). Norms, selves, and concepts. *Reason in philosophy*. Harvard University Press.
10. Brook, Andrew. (1997). *Kant and the mind*. Cambridge University Press.
11. Cali, A., Gottlob, G. and Lukasiewicz, T., 2012. A general datalog-based framework for tractable query answering over ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 14, pp.57-83.
12. Charlow, Nate. (2014). Logic and semantics for imperatives. *Journal of Philosophical Logic*, 43(4), 617-664.

13. Chellas, Brian. F. (1971). Imperatives. *Theoria*, 37(2), 114-129.
14. Coquand, T. (2010). A completeness proof for geometric logic. *Technical report, Computer Science and Engineering Department, University of Gothenburg*.
15. Dyckhoff, Roy, and Sara Negri. (2015). Geometrisation of first-order logic. *Bulletin of Symbolic Logic*, 21.2: 123-163.
16. Van Emden, Maarten H., and Robert A. Kowalski. The semantics of predicate logic as a programming language. *Journal of the ACM (JACM)* 23.4 (1976): 733-742.
17. Evans, Richard. (2017). A Kantian cognitive architecture. IACAP, 2016. To appear in *Philosophical Studies*.
18. Evans, Richard. (2017). Kant on Constituted Mental Activity. *APA on Philosophy and Computers*.
19. Friedman, Michael. (1992). *Kant and the exact sciences*. Harvard University Press.
20. Geach, Peter Thomas. (1979). Names and predicables. *Semiotics in Poland*, 240-246.
21. Gelfond, Michael, and Vladimir Lifschitz. (1988).The stable model semantics for logic programming. *ICLP/SLP*, Vol. 88.
22. Gelfond, Michael, and Vladimir Lifschitz. (1991). Classical negation in logic programs and disjunctive databases. *New generation computing*, 9(3-4), 365-385.
23. Grossi, David, and Jones, Andrew. (2013). Constitutive norms and counts-as conditionals. *Handbook of deontic logic and normative systems*, p. 407-441
24. Gurevich, Yuri, Marc Bezem, and Thierry Coquand. (2003). Newman's lemma – a case study in proof automation and geometric logic. *Bulletin of the European Association for Theoretical Computer Science*.
25. Hansen, Jörg . (2013). Imperative logic and its problems. *Handbook of Deontic Logic and Normative Systems*, 137-191.
26. Hansen, Jörg . (2008). *Imperatives and deontic logic*. PhD Thesis, Leipzig.
27. Hofstadter, Albert., and McKinsey, John. C. (1939). On the logic of imperatives. *Philosophy of Science*, 6(4), 446-457.
28. Humberstone, Lloyd. *The connectives*. MIT Press, 2011.
29. Kant, Immanuel. (1781). *Critique of pure reason*. Cambridge University Press.
30. Kant, Immanuel. (2004). *Lectures on logic*. Cambridge University Press.
31. Kaufmann, Stefan., and Schwager, Magdalena. (2009, September). A unified analysis of conditional imperatives. In *Semantics and Linguistic Theory* (Vol. 19, pp. 239-256).
32. Kitcher, Patricia. (2017). A Kantian critique of transparency. *Kant and the philosophy of mind*. Oxford University Press.
33. Kitcher, Patricia. (1993). *Kant's transcendental psychology*. Oxford University Press.
34. Kitcher, Patricia. (2011). *Kant's thinker*. Oxford University Press.
35. Kowalski, Robert, and Fariba Sadri. "An agent language with destructive assignment and model-theoretic semantics." In *International Workshop on Computational Logic in Multi-Agent Systems*, pp. 200-218. Springer, Berlin, Heidelberg, 2010.
36. Landy, David. (2015). *Kant's Inferentialism: The Case Against Hume (Vol. 11)*. Routledge.
37. Lifschitz, Vladimir, David Pearce, and Agustï¿½n Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic* (TOCL) 2.4 (2001): 526-541.
38. Longuenesse, Beatrice. (1998). *Kant and the capacity to judge*. Princeton UP.
39. Longuenesse, Beatrice. (2005). *Kant on the human standpoint*. Cambridge University Press.
40. MacFarlane, John. (2002). Frege, Kant, and the logic in logicism. *The Philosophical Review*, 111.1: 25-65.
41. Makinson, David, and Leendert Van Der Torre. (2000). Input/output logics. *Journal of Philosophical Logic*, 29.4 : 383-408.
42. Minker, Jack, and Carolina Ruiz. (1994). Semantics for disjunctive logic programs with explicit and default negation. *Fundamenta Informaticae*, 20(1, 2, 3), 145-192.
43. Mosser, Kurt. (2008). *Necessity and possibility: the logical strategy of Kant's Critique of Pure Reason*. CUA Press.
44. Negri, Sara. (2003). Contraction-free sequent calculi for geometric theories with an application to Barr's theorem. *Archive for Mathematical Logic*, 42.4: 389-401.
45. Reiter, Raymond. (1980). A logic for default reasoning. *Artificial intelligence*, 13.1-2: 81-132.
46. Ross, Alf. (1941). "Imperatives and Logic." *Theoria*, 7: 53?71.
47. Searle, John R. (1995). *The construction of social reality*. Simon and Schuster.
48. Sommers, Fred. (1983). *The logic of natural language.* Clarendon Press.
49. Stephenson, Andrew. (2013). *Kant's theory of experience* (Doctoral dissertation, University of Oxford).

50. Stephenson, Andrew. (2015). Kant on the object-dependence of intuition and hallucination. *The Philosophical Quarterly*, 65(260), 486-508.
51. Stephenson, Andrew. (2018). How to solve the knowability paradox with transcendental epistemology. *Synthese*, forthcoming.
52. Stephenson, Andrew. (2018). Logicism, possibilism, and the logic of Kantian actualism. *Critique*.
53. Strawson, Peter. (2002). *The bounds of sense*. Routledge.
54. Tiles, Mary. (2004). Kant: From general to transcendental logic. *The rise of modern logic: from Leibniz to Frege*, 3: 85-130.
55. Vranas, Peter. (2008). New foundations for imperative logic I: Logical connectives, consistency, and quantifiers. *Noûs*, 42(4), 529-572.
56. Waxman, Wayne. (2005). *Kant and the empiricists: understanding understanding*. Oxford University Press.
57. Waxman, Wayne. (2013). *Kant's anatomy of the intelligent mind*. Oxford University Press.
58. Wolff, Robert P. (1963). *Kant's theory of mental activity: a commentary on the transcendental analytic of the Critique of Pure Reason*. Harvard University Press.