Experimenting with Carmo and Jones' DDL

Lavanya Singh

September 28, 2021

Contents

theory *carmojones-DDL*
imports
Main

begin

Referencing Benzmueller, Farjami, and Parent's implementation [1]

This theory contains the axiomatization of the system and some useful abbreviations.

1 System Definition

1.1 Definitions

This section contains definitions and constants necessary to construct a DDL model.

typedecl *i* — i is the type for a set of possible worlds."

type-synonym $t = (i \Rightarrow bool)$
— t represents a set of DDL formulas.
— this set is defined by its truth function, mapping the set of worlds to the formula set's truth value.

— accessibility relations map a set of worlds to:
consts $av::i \Rightarrow t$ — actual versions of that world set
— these worlds represent what is "open to the agent"
— for example, the agent eating pizza or pasta for dinner might constitute two different actual worlds

consts $pv::i \Rightarrow t$ — possible versions of that world set
— these worlds represent was was "potentially open to the agent"
— for example, what someone across the world eats for dinner might constitute a possible world, — since the agent has no control over this

consts $ob::t \Rightarrow (t \Rightarrow bool)$ — set of propositions obligatory in this "context"
— ob(context)(term) is True if t is obligatory in the context

consts $cw::i$ — current world

1.2 Axiomatization

This subsection contains axioms. Because the embedding is semantic, these are just constraints on models.

This axiomatization comes from [2] p6 and the HOL embedding defined in Benzmuller and Parent

axiomatization where
ax-3a: $\forall w.\exists x.\ av(w)(x)$
— every world has some actual version

and *ax-4a*: $\forall w\ x.\ av(w)(x) \longrightarrow pv(w)(x)$
— all actual versions of a world are also possible versions of it

and *ax-4b*: $\forall w.\ pv(w)(w)$
— every world is a possible version of itself

and *ax-5a*: $\forall X.\neg ob(X)(\lambda w.\ \mathit{False})$
— in any arbitrary context X, something will be obligatory

and *ax-5b*: $\forall X\ Y\ Z.\ (\forall w.\ ((X(w) \wedge Y(w)) \longleftrightarrow (X(w) \wedge Z(w)))) \longrightarrow (ob(X)(Y) \longleftrightarrow ob(X)(Z))$ — note that X(w) denotes w is a member of X
— X, Y, and Z are sets of formulas
— If X ∩ Y = X ∩ Z then the context X obliges Y iff it obliges Z

— ob(X)(λ w. Fw) can be read as F ∈ ob(X)

and *ax-5c2*: $\forall X\ Y\ Z.\ (((\exists w.\ (X(w) \wedge Y(w) \wedge Z(w))) \wedge ob(X)(Y) \wedge ob(X)(Z))) \longrightarrow ob(X)(\lambda w.\ Y(w) \wedge Z(w))$

and *ax-5d*: $\forall X\ Y\ Z.\ ((\forall w.\ Y(w)\longrightarrow X(w)) \wedge ob(X)(Y) \wedge (\forall w.\ X(w)\longrightarrow Z(w)))$

$\longrightarrow ob(Z)(\lambda w.(Z(w) \wedge \neg X(w)) \vee Y(w))$
— If some subset Y of X is in ob(X) then in a larger context Z, any obligatory proposition must either be in Y or in Z-X

and *ax-5e*: $\forall X\ Y\ Z.\ ((\forall w.\ Y(w)\longrightarrow X(w)) \wedge ob(X)(Z) \wedge (\exists w.\ Y(w) \wedge Z(w))) \longrightarrow ob(Y)(Z)$
— If Z is obligatory in context X, then Z is obligatory in a subset of X called Y, if Z shares some elements with Y

1.3 Abbreviations

These abbreviations are defined in @citeBenzmullerParent p9

These are all syntactic sugar for HOL expressions, so evaluating these symbols will be light-weight

— propositional logic symbols
abbreviation *ddlneg*::$t\Rightarrow t$ (¬)
  where $\neg A \equiv \lambda w.\ \neg A(w)$
abbreviation *ddlor*::$t\Rightarrow t\Rightarrow t$ (-∨-)
  where $A \vee B \equiv \lambda w.\ (A(w) \vee B(w))$
abbreviation *ddland*::$t\Rightarrow t\Rightarrow t$ (-∧-)
  where $A\wedge B \equiv \lambda w.\ (A(w) \wedge B(w))$
abbreviation *ddlif*::$t\Rightarrow t\Rightarrow t$ (-→-)
  where $A\rightarrow B \equiv (\lambda w.\ A(w)\longrightarrow B(w))$

abbreviation *ddlequiv*::$t \Rightarrow t \Rightarrow t$ (-$\equiv$-)
  where $(A \equiv B) \equiv ((A \rightarrow B) \wedge (B \rightarrow A))$

— modal operators
abbreviation *ddlbox*::$t \Rightarrow t$ ($\Box$)
  where $\Box\ A \equiv \lambda w. \forall y.\ A(y)$
abbreviation *ddldiamond*::$t \Rightarrow t$ ($\Diamond$)
  where $\Diamond A \equiv \neg(\Box(\neg A))$

— O$\{B|A\}$ can be read as "B is obligatory in the context A"
abbreviation *ddlob*::$t \Rightarrow t \Rightarrow t$ ($O\{$-|-$\}$)
  where $O\{B|A\} \equiv \lambda\ w.\ ob(A)(B)$

— modal symbols over the actual and possible worlds relations
abbreviation *ddlboxa*::$t \Rightarrow t$ ($\Box_a$)
  where $\Box_a A \equiv \lambda x. \forall y.\ (\neg\ av(x)(y) \vee A(y))$
abbreviation *ddldiamonda*::$t \Rightarrow t$ ($\Diamond_a$)
  where $\Diamond_a A \equiv \neg(\Box_a(\neg A))$
abbreviation *ddlboxp*::$t \Rightarrow t$ ($\Box_p$)
  where $\Box_p A \equiv \lambda x. \forall y.\ (\neg\ pv(x)(y) \vee A(y))$
abbreviation *ddldiamondp*::$t \Rightarrow t$ ($\Diamond_p$)
  where $\Diamond_p A \equiv \neg(\Box_a(\neg A))$

— obligation symbols over the actual and possible worlds
abbreviation *ddloba*::$t \Rightarrow t$ (O_a)
  where $O_a\ A \equiv \lambda x.\ ob(av(x))(A) \wedge (\exists y.(av(x)(y) \wedge \neg A(y)))$
abbreviation *ddlobp*::$t \Rightarrow t$ (O_p)
  where $O_p\ A \equiv \lambda x.\ ob(pv(x))(A) \wedge (\exists y.(pv(x)(y) \wedge \neg A(y)))$

— syntactic sugar for a "monadic" obligation operator
abbreviation *ddltrue*::t ($\top$)
  where $\top \equiv \lambda w.\ True$
abbreviation *ddlob-normal*::$t \Rightarrow t$ (O $\{$-$\}$)
  where $(O\ \{A\}) \equiv (O\{A|\top\})$

— validity
abbreviation *ddlvalid*::$t \Rightarrow bool$ ($\models$-)
  where $\models A \equiv \forall w.\ A\ w$
abbreviation *ddlvalidcw*::$t \Rightarrow bool$ ($\models_c$-)
  where $\models_c A \equiv A\ cw$

1.4 Consistency

Consistency is so easy to show in Isabelle!

lemma *True* **nitpick** [*satisfy*,*user-axioms*,*show-all*,*format=2*] **oops**
— Nitpick successfully found a countermodel.
— It's not shown in the document printout, hence the oops.
— If you hover over "nitpick" in JEdit, the model will be printed to output.

```
end
```

```
theory carmojones-DDL-completeness imports carmojones-DDL
```

```
begin
```

This theory shows completeness for this logic with respect to the models presented in carmojonesDDl.thy.

2 Inference Rules

2.1 Basic Inference Rules

These inference rules are common to most modal and propostional logics

```
lemma modus-ponens: assumes ⊨ A assumes ⊨ (A → B)
  shows ⊨B
  using assms(1) assms(2) by blast
```

— Because I have not defined a “derivable” operator, inference rules are written using assumptions.

— For further meta-logical work, defining metalogical operators may be useful

```
lemma nec: assumes ⊨A shows ⊨(□A)
  by (simp add: assms)

lemma nec-a: assumes ⊨A shows ⊨(□ₐA)
  by (simp add: assms)
lemma nec-p: assumes ⊨A shows ⊨(□ₚA)
  by (simp add: assms)
```

2.2 Fancier Inference Rules

These are new rules that Carmo and Jones introduced for this logic.

```
lemma Oa-boxaO:
  assumes ⊨(B → ((¬(□((Oₐ A) → ((□ₐw) ∧ O{A|w}))))))
  shows ⊨(B → (¬(◇(Oₐ A))))
  oops
lemma Oa-boxpO:
  assumes ⊨(B → ((¬(□((Oₚ A) → ((□ₚw) ∧ O{A|w}))))))
  shows ⊨(B → (¬(◇(Oₚ A))))
  oops
```

— The oops indicates that we were not able to find a proof for these lemmas.

B and A must not contain w. not sure how to encode that requirement. one option is to define a new free variables predicate and use that, but that requires a deeper embedding than I have. May become a problem later

3 Axioms

3.1 Box

— $\Box$ is an S5 modal operator, which is where these axioms come from.
lemma *K*:
  shows $\models ((\Box(A \to B)) \to ((\Box A) \to (\Box B)))$
  by *blast*

lemma *T*:
  shows $\models ((\Box A) \to A)$
  by *blast*

lemma *5*:
  shows $\models ((\Diamond A) \to (\Box(\Diamond A)))$
  by *blast*

3.2 O

This characterization of O comes from Carmo and Jones p 593

lemma *O-diamond*:
  shows $\models (O\{A|B\} \to (\Diamond(B \wedge A)))$
  using *ax-5b ax-5a*
  by *metis*
— A is only obligatory in a context if it can possibly be true in that context.

lemma *O-C*:
  shows $\models(((\Diamond(A \wedge (B \wedge C))) \wedge (O\{B|A\} \wedge O\{C|A\})) \to (O\{B{\wedge}C|A\}))$
  by (*metis ax-5c2*)
— The conjunction of obligations in a context is obligatory in that context.
— The restriction $\Diamond(ABC)$ is to prevent contradictory obligations and contexts.

lemma *O-SA*:
  shows $\models(((\Box(A \to B)) \wedge ((\Diamond(A \wedge C)) \wedge O\{C|B\})) \to (O\{C|A\}))$
  using *ax-5e* **by** *blast*
— The principle of strengthening the antecedent.

lemma *O-REA*:
  shows $\models((\Box(A \equiv B)) \to (O\{C|A\} \equiv O\{C|B\}))$
  using *O-diamond ax-5e* **by** *blast*
— Equivalence for equivalent contexts.

lemma *O-contextual-REA*:
  shows $\models ((\Box(C \to (A \equiv B))) \to (O\{A|C\} \equiv O\{B|C\}))$
  by (*metis ax-5b*)
— The above lemma, but in some context C.

lemma *O-nec*:
  shows $\models(O\{B|A\} \to (\Box O\{B|A\}))$

```
  by simp
```

— Obligations are necessarily obligated.

```
lemma ax-5b'':
  shows ob X Y ⟷ ob X (λz. (Y z) ∧ (X z))
  by (metis (no-types, lifting) ax-5b)
```

```
lemma O-to-O:
  shows ⊨(O{B|A}→O{(A→B)|⊤})
proof−
  have ∀X Y Z. (ob X Y ∧ (∀w. X w ⟶ Z w)) ⟶ ob Z (λw.(Z w ∧ ¬X w) ∨ Y w)
  by (smt ax-5d ax-5b ax-5b'')
  thus ?thesis
  proof −
     have f1: ∀p pa pb. ((¬ (ob p pa)) ∨ (∃i. (p∧(¬ pb)) i)) ∨ (ob pb ((pb∧(¬ p))∨  pa))
      using ‹∀X Y Z. ob X Y ∧ (⊨(X→Z)) ⟶ ob Z ( (Z∧(¬ X))∨ Y)› by force
    obtain ii :: (i ⇒ bool) ⇒ (i ⇒ bool) ⇒ i where
      ∀x0 x2. (∃v3. (x2∧(¬ x0)) v3) = (x2∧(¬ x0)) (ii x0 x2)
      by moura
    then have ∀p pa pb. ((¬ ob p pa) ∨ (p∧(¬ pb)) (ii pb p)) ∨ ob pb ( (pb∧(¬ p))∨ pa)
      using f1 by presburger
    then show ?thesis
      by fastforce
  qed
qed
```

— Moving from the dyadic to monadic obligation operators.

3.3 Possible Box

— $\Box_p$ is a KT modal operator.

```
lemma K-boxp:
  shows ⊨((□p(A → B)) → ((□pA) → (□pB)))
  by blast
lemma T-boxp:
  shows ⊨((□pA) → A)
  using ax-4b by blast
```

3.4 Actual Box

— $\Box_a$ is a KD modal operator.

```
lemma K-boxa:
  shows ⊨((□a(A → B)) → ((□aA) → (□aB)))
  by blast
lemma D-boxa:
  shows ⊨((□aA) → (◇aA))
  using ax-3a by blast
```

3.5 Relations Between the Modal Operators

— Relation between $\Box$, $\Box_a$, and $\Box_p$.
lemma *box-boxp*:
shows $\models((\Box A) \rightarrow (\Box_p A))$
by *auto*
lemma *boxp-boxa*:
shows $\models((\Box_p A) \rightarrow (\Box_a A))$
using *ax-4a* **by** *blast*

— Relation between actual and possible O and $\Box$.
lemma *not-Oa*:
shows $\models((\Box_a A) \rightarrow ((\neg(O_a\ A)) \wedge (\neg(O_a\ (\neg A)))))$
using *O-diamond* **by** *blast*
lemma *not-Op*:
shows $\models((\Box_p A) \rightarrow ((\neg(O_p\ A)) \wedge (\neg(O_p\ (\neg A)))))$
using *O-diamond* **by** *blast*
lemma *equiv-Oa*:
shows $\models((\Box_a(A \equiv B)) \rightarrow ((O_a\ A) \equiv (O_a\ B)\ ))$
using *O-contextual-REA* **by** *blast*
lemma *equiv-Op*:
shows $\models((\Box_p(A \equiv B)) \rightarrow ((O_p\ A) \equiv (O_p\ B)\ ))$
using *O-contextual-REA* **by** *blast*

— relationships between actual and possible O and $\Box$ and O proper.
lemma *factual-detach-a*:
shows $\models(((O\{B|A\} \wedge (\Box_a A)) \wedge ((\Diamond_a B) \wedge (\Diamond_a(\neg B)))) \rightarrow (O_a\ B))$
using *O-SA* **by** *auto*
lemma *factual-detach-p*:
shows $\models(((O\{B|A\} \wedge (\Box_p A)) \wedge ((\Diamond_p B) \wedge (\Diamond_p(\neg B)))) \rightarrow (O_p\ B))$
by (*smt O-SA boxp-boxa*)

end

theory *categorical-imperative-1* **imports** *carmojones-DDL-completeness*

begin

4 The Categorical Imperative

4.1 Simple Formulation of the Kingdom of Ends

This is my first attempt at formalizing the concept of the Kingdom of Ends

NOTE: this attempt revealed a bug in my embedding. I've included it as an artifact, but none of these theorems hold anymore (hence the oops).

abbreviation *ddlpermissable*::$t \Rightarrow t$ (P-)
  where $(P\ A) \equiv (\neg(O\ \{\neg A\}))$
— This operator represents permissibility
— Will be useful when discussing the categorical imperative
— Something is permissible if it is not prohibited
— Something is prohibited if its negation is obligatory

lemma *kingdom-of-ends-1*:
  shows $\models ((O\ \{A\}) \rightarrow (\Box\ (P\ A)))$
  oops
— One interpretation of the categorical imperative is that something is obligatory only if it is permissible in every ideal world
— This formulation mirrors the kingdom of ends.
— This formulation is already a theorem of carmo and jones' DDL!.
— It can be shown using the O diamond rule, which just says that obligatory things must be possible.
— There are two possibilities: either the logic is already quite powerful OR this formulation is "empty".

lemma *kingdom-of-ends-2*:
  shows $\models ((\Box\ (P\ A)) \rightarrow (O\ \{A\}))$
  oops

— Notice also that ideally, this relationship does not hold in the reverse direction.
— Plenty of things are necessarily permissible (drinking water) but not obligatory.
— Very strange that this is a theorem in this logic.....
— That being said, Isabelle seems quite upset with this proof and is very slow to resconstruct it
— I am struggling to recreate this proof on paper

lemma *permissible-to-ob*:
  shows $\models ((P\ A) \rightarrow (O\ \{A\}))$
  oops

— Uh-oh.....this shouldn't be true...
— Not all permissable things are obligatory.....

lemma *weaker-permissible-to-ob*:
shows $\models ((\Diamond\ (P\ A)) \rightarrow O\ \{A\})$
oops
— Makes sense that this follows from the reverse kingdom of ends.
— Obligation and necessity/possibility are separated in this logic
— Both the dyadic obligation and necessity operator are world agnostic

lemma *contradictory-obligations*:
shows $\models(\neg\ ((O\ \{A\}) \wedge (O\ \{\neg\ A\})))$
nitpick[*user-axioms*]
oops
— What is the cause of the above strangeness?
— This very intuitive theorem holds in my logic but not in BFP's
— It's clear that this theorem results in the strange results above.
— Conclusion: There is a bug in my embedding
— Nitpick found a counterexample for card i = 2:
Free variable: A = $(\lambda x.\_)(i_1 :=$ False, $i_2 :=$ True)

Sidebar: the above theorem is really intuitive - it seems like we wouldn't want contradictory things to be obligatory in any logic. But for some reason, not only is it not a theorem of Carmo and Jones' logic, it actually implies some strange conclusions, including that everything is either permissible or obligatory. It's not clear to me from a semantic perspective why this would be the case. In fact this theorem seems like a desirable property. Potential avenue for exploration

Did some debugging. What was the problem? A misplaced parentheses in the definition of ax5b that led to a term being on the wrong side of an implication. Computer Science :(

After the debugging, all of this is no longer true! On to the next attempt :)

end

theory *categorical-imperative-naive* **imports** *carmojones-DDL-completeness*

begin

5 The Categorical Imperative

5.1 Simple Formulation of the Formula of Universal Law

This is my second attempt at formalizing the Formula of Universal Law

abbreviation *ddlpermissable*::$t \Rightarrow t$ (*P*-)
  where $(P\ A) \equiv (\neg(O\ \{\neg A\}))$
— This operator represents permissibility
— Will be useful when discussing the categorical imperative
— Something is permissible if it is not prohibited
— Something is prohibited if its negation is obligatory

Let's consider a naive reading of the Formula of Universal Law (FUL). From the Groundwork, 'act only in accordance with that maxim through which you can at the same time will that it become a universal law'. What does this mean in DDL? One interpretation is if A is not necessarily permissible, then its negation is obligated.

axiomatization where
FUL-1: $\models ((\neg(\Box\ (P\ A))) \rightarrow (O\ \{(\neg A)\}))$

5.2 Basic Tests

lemma *True* **nitpick** [*satisfy*,*user-axioms*,*format=2*] **oops**
— "Nitpick found a model for card i = 1:
Empty assignment"
— Nitpick tells us that the FUL is consistent
— "oops" after Nitpick does not mean Nitpick failed.

lemma *something-is-obligatory*:
  shows $\forall\ w.\ \exists\ A.\ O\ \{A\}\ w$
  nitpick [*user-axioms*]
  oops
— We might think that in every world we want something to be obligated.
— Sadly, Sledgehammer times out trying to prove this. Let's relax this
— "Nitpick found a counterexample for card i = 1:
Empty assignment"
— Nitpick to the rescue! The formula is in fact not valid.

lemma *something-is-obligatory-2*:
  shows $\forall\ w.\ \exists\ A.\ O\ \{A\}\ w$
  nitpick [*user-axioms*, *falsify=false*]
  oops
— "Nitpick found a model for card i = 1:
Skolem constant: A = $(\lambda x.\_)(i_1 :=$ True)"

— Nitpick tells us that the formula is consistent - it found a model where the formula is true.
— This means that our model is underspecified - this formula is neither valid nor inconsistent.

lemma *something-is-obligatory-relaxed*:
  shows $\exists\ A\ w.\ O\ \{A\}\ w$
  nitpick [*user-axioms*]
  oops
— "Nitpick found a counterexample for card i = 1:
Empty assignment"
— The relaxed version definitely isn't valid.

lemma *something-is-obligatory-relaxed-2*:
  shows $\exists\ A\ w.\ O\ \{A\}\ w$
  nitpick [*user-axioms*, *falsify=false*]
  oops
— "Nitpick found a model for card i = 1:
Skolem constant: A = $(\lambda x.\_)(i_1$:= True)"
— Nitpick tells us that the formula is consistent - it found a model where the formula is true.
— The model seems underspecified.

5.3 Specifying the Model

Let's specify the model. What if we add something impermissible?

consts *M*::*t*
abbreviation *murder-wrong*::*bool* **where** *murder-wrong* $\equiv \models(O\ \{\neg\ M\})$

lemma *something-is-obligatory-2*:
  assumes *murder-wrong*
  shows $\forall\ w.\ \exists\ A.\ O\ \{A\}\ w$
  using *assms* **by** *auto*
— It works this time, but I think "murder wrong" might be too strong of an assumption

abbreviation *poss-murder-wrong*::*bool* **where** *poss-murder-wrong* $\equiv \models(\Diamond\ (O\ \{\neg\ M\}))$

lemma *wrong-if-posibly-wrong*:
  assumes *poss-murder-wrong*
  shows *murder-wrong*
  using *assms* **by** *blast*
— This lemma holds and uses a slightly weaker assumption. This also seems to get closer to the "heart" of this naive interpretation. We really want to say that if something isn't necessarily obligated, it's not obligated anywhere."

Let's try an even weaker assumption: Not everyone can lie.

typedecl *person*

consts *lies*::*person*⇒*t*
consts *me*::*person*

lemma *breaking-promises*:
  assumes ¬ (∀ *x*. *lie*(*x*) *cw*) ∧ (*lie*(*me*) *cw*)
  shows (*O* {¬ (*lie*(*me*))}) *cw*
  nitpick [*user-axioms*]
  oops
— No proof found. Makes sense:
— This version of FUL simply universalizes across worlds (using the □ operator),
— But NOT across people, which is really what the most obvious reading of FUL implies
— "Nitpick found a counterexample for card person = 2 and card i = 2:
Free variable: lie = $(\lambda x.\_)(p_1 := (\lambda x.\_)(i_1 :=$ True, $i_2 :=$ False), $p_2 := (\lambda x.\_)(i_1 :=$ False, $i_2 :=$ False))"

lemma *universalizability*:
  assumes ⊨ *O* {(*lie*(*me*))}
  shows ∀ *x*. ⊨ (*O* {(*lie*(*x*))})
  nitpick [*user-axioms*] **oops**
— Nitpick found a counterexample for card person = 2 and card i = 2:
Free variable: lie = $(\lambda x.\_)(p_1 := (\lambda x.\_)(i_1 :=$ False, $i_2 :=$ True), $p_2 := (\lambda x.\_)(i_1 :=$ False, $i_2 :=$ False)) Skolem constant: x = p_2
— This lemma demonstrates the point even more clearly - we really want to think that obligations are consistent across people, but because we don't have a notion of agency, we don't have that property.

5.4 Consistent Sentences

The above section tested validity. We might also be interested in some weaker properties

Let's test whether certain sentences are consistent - can we find a model that makes them true?

lemma *permissible*:
  fixes *A*
  shows ((¬ (*O* {*A*})) ∧ (¬ (*O* {¬ *A*}))) *w*
  nitpick [*user-axioms*, *falsify=false*] **oops**
— "Nitpick found a model for card i = 1:
Free variable: A = $(\lambda x.\_)(i_1 :=$ False)"
— Awesome! Permissible things are consistent - clearly we've fixed the bug from categorical_imperative_1
— Note that apparently it's not clear @cite kitcher if Kant actually thought that permissibility was a coherent concept. Either way, I think permissibility is a pretty widely accepted ethical phenomenon.

lemma *conflicting-obligations*:
  fixes *A*
  shows (*O* {*A*} ∧ *O* {¬ *A*}) *w*

nitpick [*user-axioms*, *falsify=false*] **oops**
— “Nitpick found a model for card i = 2:
Free variable: A = $(\lambda x.\_)(i_1$:= False, i_2 := True)”
— Oh no! Nitpick found a model with conflicting obligations - that’s bad!

5.5 Metaethical Tests

lemma *FUL-alternate*:
shows $\models ((\Diamond\ (O\ \{\neg\ A\})) \rightarrow (O\ \{\neg\ A\}))$
by *simp*
— One problem becomes obvious if we look at the definition of permissible
— Expanding the FUL gives us: $\sim \Box \sim O(\sim A) \longrightarrow O(\sim A)$
— By modal duals we get that $\diamond O(\sim A) \longrightarrow O(\sim A)$
— This means that if something is possibly prohibited, it is in fact prohibited.
— I’m not convinced that this is a desirable property of an ethical theory.

lemma *arbitrary-obligations*:
fixes $A::t$
shows $O\ \{A\}\ w$
nitpick [*user-axioms=true*] **oops**
— “Nitpick found a counterexample for card i = 1:
Free variable: A = $(\lambda x.\_)(i_1$:= False)”
— This is good! Shows us that any arbitrary term isn’t obligatory.

lemma *removing-conflicting-obligations*:
assumes $\forall A.\ \models (\neg\ (O\ \{A\}\ \wedge\ O\ \{\neg\ A\}))$
shows *True*
nitpick [*satisfy*,*user-axioms*,*format=2*] **oops**
— “ Nitpick found a model for card i = 1:
Empty assignment”
— We can disallow conflicting obligations and the system is still consistent - that’s good.

lemma *implied-contradiction*:
fixes $A::t$
fixes $B::t$
assumes $\models(\neg\ (A\ \wedge\ B))$
shows $\models(\neg\ (O\ \{A\}\ \wedge\ O\ \{B\}))$
nitpick [*user-axioms*]
proof −
have $\models(\neg(\Diamond(A\ \wedge\ B)))$
by (*simp add*: *assms*)
then have $\models(\neg\ (O\ \{A \wedge B\}))$ **by** (*smt carmojones-DDL-completeness.O-diamond*)
thus *?thesis* **oops**
— [4] mentions that if two maxims imply a contradiction, they must not be willed.
— Above is a natural interpretation of this fact that we are, so far, unable to prove.
— “Nitpick found a counterexample for card i = 2:
Free variables: A = $(\lambda x.\_)(i_1$:= True, i_2 := False) B = $(\lambda x.\_)(i_1$:= False, i_2 := True)”

— This isn't actually a theorem of the logic as formed - clearly this is a problem.

lemma *distribute-obligations-if*:
  assumes $\models O \{A \wedge B\}$
  shows $\models (O \{A\} \wedge O \{B\})$
  nitpick [*user-axioms*, *falsify=true*, *verbose*]
  oops

— Nitpick can't find a countermodel for this theorem, and sledgehammer can't find a proof.

— Super strange. I wonder if this is similar to $\Box(A \wedge B)$ vs $\Box A \wedge \Box B$

lemma *distribute-boxes*:
  assumes $\models(\Box(A \wedge B))$
  shows $\models ((\Box A) \wedge (\Box B))$
  using *assms* **by** *blast*

— We really expect the O operator to be acting like the □ operator. It's like a modal necessity operator, like necessity across ideal worlds instead of actual worlds. Therefore, we'd expect theorems that hold of □ to also hold of O.

lemma *distribute-obligations-onlyif*:
  assumes $\models (O \{A\} \wedge O \{B\})$
  shows $\models O \{A \wedge B\}$
  nitpick [*user-axioms*] **oops**

— "Nitpick found a counterexample for card i = 2:
Free variables: A = $(\lambda x.\_)(i_1$:= True, i_2 := False) B = $(\lambda x.\_)(i_1$:= False, i_2 := True)"

— If this was a theorem, then contradictory obligations would be ruled out pretty immediately.

— Note that all of this holds in CJ's original DDL as well, not just my modified version.

— We might imagine adding this equivalence to our system.

lemma *ought-implies-can*:
  shows $\forall A. \models (O \{A\} \rightarrow (\Diamond A))$
  using *O-diamond* **by** *blast*

— "ought implies can" is often attributed to Kant and is a pretty basic principle - you can't be obligated to do the impossible. I'm not surprised that our base logic has this as an axiom. It's often said to be the central motivation behind deontic logics.

end

theory *kroy*
  imports *carmojones-DDL*

begin

This theory will contain a formalization of the CI based on Moshe Kroy's partial formalization. [5]

6 Kroy's Formalization of the Categorical Imperative

6.1 The Substitution Operator

6.1.1 Definition

typedecl *s* — s is the type for a "subject", like the subject of a sentence
— Intuitively, we need some notion of "x does action", which we can write as "x is the subject of the sentence 'does action'"

type-synonym $os = (s \Rightarrow t)$
— An open sentence is a generalized version of Kroy's substitution operator [5] 196
— "does action" is an open sentence that can be instantiated with a subject
— "P sub (d/e)" can be written as "S(e)", where S(d) = P
— So the terms that we substitute into are actually instantiations of an open sentence, and substitution just requires re-instantiating the open sentence with a different subject

6.1.2 Abbreviations

abbreviation *os-neg*::*os* ⇒ *os* (¬-)
  where $(\neg A) \equiv \lambda x.\ \neg(A(x))$
abbreviation *os-and*::*os* ⇒ *os* ⇒ *os* (-∧-)
  where $(A \wedge B) \equiv \lambda x.\ ((A(x)) \wedge (B(x)))$
abbreviation *os-or*::*os* ⇒ *os* ⇒ *os* (-∨-)
  where $(A \vee B) \equiv \lambda x.\ ((A(x)) \vee (B(x)))$
abbreviation *os-ob*::*os* ⇒ *os* (O{-})
  where $\text{O}\{A\} \equiv \lambda x.\ (O\ \{A(x)\})$
— We could probably do without these abbreviations, but they will simplify the notiation a bit and unify it with Kroy's original paper.

abbreviation *ddl-permissible*::*t*⇒*t* (*P* {-})
  where $P\ \{A\} \equiv \neg\ (O\ \{\neg\ A\})$
abbreviation *os-permissible*::*os*⇒*os* (P {-})
  where $\text{P}\ \{A\} \equiv \lambda x.\ P\ \{A(x)\}$
— Carmo and Jones don't make much use of permissibility, but we will find it useful here.

6.2 Differences Between Kroy's Logic (Kr) and DDL

[5] uses a different logic than DDL. Let's see if the semantics that Kr requires hold in DDL

lemma *permissible-semantic-faithful*:
fixes A w
shows $P\ \{A\}\ w \longrightarrow (\exists\, x.\ A(x))$
nitpick[*user-axioms*] **oops**
— The most faithful interpretation of Kr is that if A is permissible in a context, then it must be true at some world in that context. Kr operates under the "deontic alternatives" view, summarized by Solt as "A proposition of the sort OA is true at the actual world w if and only if A is true at every deontic alternative world to t." Under this view, permissible propositions are obligated at some deontic alternative, but not at all of them.
DDL does not adopt a deontic alternatives view, which is why this proposition seems wildly counterintuitive in DDL. In DDL, the ob operator abstracts away the notion of deontic alternatives and completely determines obligations. Even if one belives that permissible statements should be true at some deontic alternative, it's not clear that permissible statements must be realized at some world, hence the failure of this lemma in DDL.
— Nitpick found a counterexample for card i = 1:
Free variable: A = $(\lambda x.\_)(i_1 :=$ False)

lemma *permissible-semantic-faithful-2*:
fixes A w
shows $P\ \{A\}\ w \longrightarrow (\exists\, x.\ O\ \{A\}\ x)$
nitpick[*user-axioms*] **oops**
— Nitpick found a counterexample for card i = 1:
Free variable: A = $(\lambda x.\_)(i_1 :=$ False)
— This is the most clear lemma that we would expect to hold under the deontic alternatives view. The fact that it doesn't shows DDL is not a logic of deontic alternatives. There are pros and cons to this approach. The deontic alternatives view is quite simple to visualize and offers clear intuition. On the other hand, DDL's ob function can encode more complex relations than the deontic alternatives view, and can encode these in a more intuitive manner. The notion of a "deontically perfect alternative" is a squishy one, and an ob function more directly captures the idea of obligation.

lemma *permissible-semantic-vacuous*:
fixes A w
shows $P\ \{A\}\ w \longrightarrow (\exists\, x.\ ob(x)(A))$
nitpick[*user-axioms*] **oops**
— Kr does not allow vasuously permissible statements -¿ if something is permissible it has to be obligated at some deontically perfect alternative. In DDL, this can be roughly translated as, if A is permissible, it is obligated in some context.
— Nitpick found a counterexample for card i = 1:
Free variable: A = $(\lambda x.\_)(i_1 :=$ False)
— In order to make this true, we'd have to require that everything is either oblig-

atory or prohibited somewhere. But as found in the buggy version of DDL, that breaks everything and destroys the notion of permissibility everywhere. I am going to allow for vacuous permissibility. If something breaks later in Kr, it may be because of this.

```
lemma permissible-ob:
  fixes A w
  shows O {A} w ⟶ P {A} w
  nitpick [user-axioms] oops
```

— Nitpick found a counterexample for card i = 2:
Free variable: A = ($\lambda x.\_$)(i_1 := False, i_2 := True)
— This one is definitely problematic. Being permissible should be a precondition for being obligatory. In my eventual logic, I will need to add this as an axiom, because I can't see any ethical theory succeeding without this.

6.3 The Categorical Imperative

```
abbreviation FUL::bool where FUL ≡ ∀ w A. ((∃ p. (P {A}p )w) ⟶ (∀ x. (P
{A}x)w))
```

— This is Kroy's formalization of the FUL in DDL. Recall that the FUL says "act only in accordance with that maxim through which you can at the same time will that it become a universal law" [3] Kroy interprets this to mean that if an action A is permissible for some agent p, then it must be permissible for everyone. This formalizes the important moral intuition that the formula of universal law prohibits free-riding. No one is a moral exception

```
lemma FUL:
  shows FUL
  nitpick[user-axioms] oops
```

— Nitpick found a counterexample for card s = 2 and card i = 2:
Skolem constants: A = ($\lambda x.\_$)(s_1 := ($\lambda x.\_$)(i_1 := True, i_2 := True), s_2 := ($\lambda x.\_$)(i_1 := False, i_2 := False)) p = s_1 x = s_2 This formalization doesn't hold in DDL. Good - this means that adding it as an axiom will change the logic.

```
lemma ⊨ (¬ (O {¬ ⊤}))
  by (simp add: ax-5a)

lemma complete:
  shows (⊨ (((A → ¬⊤)))) ⟶ (⊨ (¬ (O {A})))
proof –
  have (∃ x. (¬ (◇ A)) x) ⟶ (∃ x. (¬ O {A}) x)
    by (simp add: ax-5a ax-5b)
  thus ?thesis
    by blast
qed

axiomatization where CI: CI and
possible: (∀ w. ¬ ◇A w) ⟶ ⊨(O {¬ A})
```

— We really need a way to add negative obligations - to related the concept of contradiction and obligation. This seems reasonable - if A is never possible at any world, then it's prohibited.

```
and hmm: ⊨ (¬ (O {¬ ⊤}))
```

6.4 Tests

```
lemma True nitpick [satisfy,user-axioms] oops
— Nitpick found a model for card s = 1 and card i = 1:
Empty assignment
— The categorical imperative is consistent!
```

6.4.1 Specifying the Model

```
lemma breaking-promises:
  fixes me::s
  fixes lie::os
  assumes ∃ x. (¬ (◇(lie(x))) cw)
  shows ∃ x. (¬ (O {lie(x)})) cw
  by (metis assms ax-5a ax-5b)
```

```
lemma breaking-promises:
  fixes me::s
  fixes lie::os
  assumes ∃ x. (¬ (◇(lie(x))) cw)
  shows ∃ x. (¬ (P {lie(x)})) cw
  nitpick[user-axioms] oops
```

6.4.2 Metaethical Tests

6.4.3 Kroy's Tests

```
end
```

References

[1] C. Benzmüller, A. Farjami, and X. Parent. Dyadic deontic logic in hol: Faithful embedding and meta-theoretical experiments. In M. Armgardt, H. C. Nordtveit Kvernenes, and S. Rahman, editors, *New Developments in Legal Reasoning and Logic: From Ancient Law to Modern Legal Systems*, volume 23 of *Logic, Argumentation & Reasoning*. Springer Nature Switzerland AG, 2021.

[2] J. Carmo and A. Jones. Completeness and decidability results for a logic of contrary-to-duty conditionals. *J. Log. Comput.*, 23:585–626, 2013.

[3] I. Kant. *Groundwork of the Metaphysics of Morals*. Cambridge University Press, Cambridge, 1785.

[4] C. Korsgaard. Kant's Formula of Universal Law. *Pacific Philosophical Quarterly*, 66:24–47, 1985.

[5] M. Kroy. A partial formalization of kant's categorical imperative. an application of deontic logic to classical moral philosophy. *Kant-Studien*, 67(1-4):192–209, 1976.