Experimenting with Carmo and Jones' DDL

Lavanya Singh

March 31, 2021

Contents

theory *carmojones-DDL*
imports
Main

begin

Referencing Benzmuller 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*::$t \Rightarrow t$ ($\neg$)
  where $\neg A \equiv \lambda w.\ \neg A(w)$
abbreviation *ddlor*::$t \Rightarrow t \Rightarrow t$ (-$\vee$-)
  where $A \vee B \equiv \lambda w.\ (A(w) \vee B(w))$
abbreviation *ddland*::$t \Rightarrow t \Rightarrow t$ (-$\wedge$-)
  where $A \wedge B \equiv \lambda w.\ (A(w) \wedge B(w))$
abbreviation *ddlif*::$t \Rightarrow t \Rightarrow t$ (-$\rightarrow$-)
  where $A \rightarrow B \equiv (\lambda w.\ A(w) \longrightarrow B(w))$

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

— 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*] ⟨*proof*⟩

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** $\models A$ **assumes** $\models (A \rightarrow B)$
shows $\models B$
⟨*proof*⟩

lemma *nec*: **assumes** $\models A$ **shows** $\models (\Box A)$
⟨*proof*⟩

lemma *nec-a*: **assumes** $\models A$ **shows** $\models (\Box_a A)$
⟨*proof*⟩
lemma *nec-p*: **assumes** $\models A$ **shows** $\models (\Box_p A)$
⟨*proof*⟩

2.2 Fancier Inference Rules

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

lemma *Oa-boxaO*:
assumes $\models (B \rightarrow ((\neg(\Box((O_a\ A) \rightarrow ((\Box_a w) \wedge O\{A|w\}))))))$
shows $\models (B \rightarrow (\neg(\Diamond(O_a\ A))))$
⟨*proof*⟩
lemma *Oa-boxpO*:
assumes $\models (B \rightarrow ((\neg(\Box((O_p\ A) \rightarrow ((\Box_p w) \wedge O\{A|w\}))))))$
shows $\models (B \rightarrow (\neg(\Diamond(O_p\ A))))$
⟨*proof*⟩

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. If Benzmuller and Parent can survive without these inference rules, so can I

3 Axioms

3.1 Box

— $\Box$ is an S5 modal operator, which is where these axioms come from.

lemma *K*:
shows $\models ((\Box(A \to B)) \to ((\Box A) \to (\Box B)))$
$\langle proof \rangle$

lemma *T*:
shows $\models ((\Box A) \to A)$
$\langle proof \rangle$

lemma *5*:
shows $\models ((\Diamond A) \to (\Box(\Diamond A)))$
$\langle proof \rangle$

3.2 O

This characterization of O comes from Carmo and Jones p 593

lemma *O-diamond*:
shows $\models (O\{A|B\} \to (\Diamond(B \wedge A)))$
$\langle proof \rangle$

lemma *O-C*:
shows $\models(((\Diamond(A \wedge (B \wedge C))) \wedge (O\{B|A\} \wedge O\{C|A\})) \to (O\{B{\wedge}C|A\}))$
$\langle proof \rangle$

lemma *O-SA*:
shows $\models(((\Box(A \to B)) \wedge ((\Diamond(A \wedge C)) \wedge O\{C|B\})) \to (O\{C|A\}))$
$\langle proof \rangle$

lemma *O-REA*:
shows $\models((\Box(A \equiv B)) \to (O\{C|A\} \equiv O\{C|B\}))$
$\langle proof \rangle$

lemma *O-contextual-REA*:
shows $\models ((\Box(C \to (A \equiv B))) \to (O\{A|C\} \equiv O\{B|C\}))$
$\langle proof \rangle$

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

lemma *ax-5b''*:
shows $ob\ X\ Y \longleftrightarrow ob\ X\ (\lambda z.\ (Y\ z) \wedge (X\ z))$
$\langle proof \rangle$

lemma *O-to-O*:
shows $\models(O\{B|A\} \to O\{(A \to B)|\top\})$
$\langle proof \rangle$

3.3 Possible Box

— $\Box_p$ is a KT modal operator.
lemma *K-boxp*:
shows $\models((\Box_p(A \rightarrow B)) \rightarrow ((\Box_p A) \rightarrow (\Box_p B)))$
⟨*proof*⟩
lemma *T-boxp*:
shows $\models((\Box_p A) \rightarrow A)$
⟨*proof*⟩

3.4 Actual Box

— $\Box_a$ is a KD modal operator.
lemma *K-boxa*:
shows $\models((\Box_a(A \rightarrow B)) \rightarrow ((\Box_a A) \rightarrow (\Box_a B)))$
⟨*proof*⟩
lemma *D-boxa*:
shows $\models((\Box_a A) \rightarrow (\Diamond_a A))$
⟨*proof*⟩

3.5 Relations Between the Modal Operators

— Relation between $\Box$, $\Box_a$, and $\Box_p$.
lemma *box-boxp*:
shows $\models((\Box A) \rightarrow (\Box_p A))$
⟨*proof*⟩
lemma *boxp-boxa*:
shows $\models((\Box_p A) \rightarrow (\Box_a A))$
⟨*proof*⟩
lemma *not-Oa*:
shows $\models((\Box_a A) \rightarrow ((\neg(O_a\ A)) \wedge (\neg(O_a\ (\neg A)))))$
⟨*proof*⟩
lemma *not-Op*:
shows $\models((\Box_p A) \rightarrow ((\neg(O_p\ A)) \wedge (\neg(O_p\ (\neg A)))))$
⟨*proof*⟩
lemma *equiv-Oa*:
shows $\models((\Box_a(A \equiv B)) \rightarrow ((O_a\ A) \equiv (O_a\ B)\ ))$
⟨*proof*⟩
lemma *equiv-Op*:
shows $\models((\Box_p(A \equiv B)) \rightarrow ((O_p\ A) \equiv (O_p\ B)\ ))$
⟨*proof*⟩
lemma *factual-detach-a*:
shows $\models(((O\{B|A\} \wedge (\Box_a A)) \wedge ((\Diamond_a B) \wedge (\Diamond_a(\neg B)))) \rightarrow (O_a\ B))$
⟨*proof*⟩
lemma *factual-detach-p*:
shows $\models(((O\{B|A\} \wedge (\Box_p A)) \wedge ((\Diamond_p B) \wedge (\Diamond_p(\neg B)))) \rightarrow (O_p\ B))$
⟨*proof*⟩

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 $\models ((O\ \{A\}) \rightarrow (\Box\ (P\ A)))$
  ⟨*proof*⟩

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

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

lemma *weaker-permissible-to-ob*:
  shows $\models ((\Diamond\ (P\ A)) \rightarrow O\ \{A\})$
    ⟨*proof*⟩

lemma *contradictory-obligations*:
  shows $\models (\neg\ ((O\ \{A\}) \wedge (O\ \{\neg\ A\})))$
  ⟨*proof*⟩

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*] ⟨*proof*⟩

lemma *something-is-obligatory*:
 shows $\forall\ w.\ \exists\ A.\ O\ \{A\}\ w$
 nitpick [*user-axioms*]
 ⟨*proof*⟩

lemma *something-is-obligatory-2*:
 shows $\forall\ w.\ \exists\ A.\ O\ \{A\}\ w$
 nitpick [*user-axioms*, *falsify=false*]
 ⟨*proof*⟩

lemma *something-is-obligatory-relaxed*:
 shows $\exists\ A\ w.\ O\ \{A\}\ w$
 nitpick [*user-axioms*]
 ⟨*proof*⟩

lemma *something-is-obligatory-relaxed-2*:
 shows $\exists\ A\ w.\ O\ \{A\}\ w$
 nitpick [*user-axioms*, *falsify=false*]
 ⟨*proof*⟩

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 $\forall\ w.\ \exists\ A.\ O\ \{A\}\ w$
  $\langle proof \rangle$

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*
  $\langle proof \rangle$

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

typedecl *person*
consts $lies::person \Rightarrow t$
consts $me::person$

lemma *breaking-promises*:
  assumes $\neg\ (\forall x.\ lie(x)\ cw) \wedge (lie(me)\ cw)$
  shows $(O\ \{\neg\ (lie(me))\})\ cw$
  nitpick [*user-axioms*]
  $\langle proof \rangle$

lemma *universalizability*:
  assumes $\models\ O\ \{(lie(me))\}$
  shows $\forall x.\ \models\ (O\ \{(lie(x))\})$
  nitpick [*user-axioms*] $\langle proof \rangle$

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 $((\neg\ (O\ \{A\})) \wedge (\neg\ (O\ \{\neg\ A\})))\ w$
  nitpick [*user-axioms*, *falsify=false*] $\langle proof \rangle$

lemma *conflicting-obligations*:

fixes A
shows $(O\ \{A\} \wedge O\ \{\neg\ A\})\ w$
nitpick [*user-axioms*, *falsify=false*] ⟨*proof*⟩

5.5 Metaethical Tests

lemma *FUL-alternate*:
shows $\models ((\Diamond\ (O\ \{\neg\ A\})) \rightarrow (O\ \{\neg\ A\}))$
⟨*proof*⟩

lemma *arbitrary-obligations*:
fixes $A{::}t$
shows $O\ \{A\}\ w$
nitpick [*user-axioms=true*] ⟨*proof*⟩

lemma *removing-conflicting-obligations*:
assumes $\forall A.\ \models (\neg\ (O\ \{A\} \wedge O\ \{\neg\ A\}))$
shows *True*
nitpick [*satisfy*,*user-axioms*,*format=2*] ⟨*proof*⟩

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*⟩

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

lemma *distribute-boxes*:
assumes $\models(\ \Box(A \wedge B))$
shows $\models ((\Box A) \wedge (\Box B))$
⟨*proof*⟩

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

lemma *ought-implies-can*:
shows $\forall A.\ \models (O\ \{A\} \rightarrow (\Diamond A))$
⟨*proof*⟩

end

References

[1] C. Benzmüller, A. Farjami, and X. Parent. Faithful semantical embedding of a dyadic deontic logic in HOL. *CoRR*, abs/1802.08454, 2018.

[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.