Semantics for Conditional Logics: A First-Order Formalization with Applications

Branden Fitelson
November 17, 2008

1 The Formalization

Let Rpw_1w_2 be a three-place relation between a proposition p and a pair of worlds w_1 and w_2. Rpw_1w_2 will be true iff $w_2 \in f_p(w_1)$. Let Tpw be a two-place relation between a proposition p and a world w. Tpw will be true iff $w \in [p]$. With these two relations, we can formalize the semantics for conditional logics. Next, we will give the basic, underlying semantical definitions involving T and R. Here, the quantifiers will be meta-theoretic, as will the connectives $\Rightarrow$ (meta-theoretic conditional) $\sim$ (meta-theoretic negation), & (meta-theoretic conjunction), $\curlyvee$ (meta-theoretic disjunction), and $\Leftrightarrow$ (meta-theoretic biconditional). As usual, the meta-theory will be *classical*. Before giving the semantical definitions involving T and R, we need to add two (meta–theoretic) monadic predicates Px (x is a proposition) and Wx (x is a world) to be able to distinguish worlds and propositions. And, we need to add several typing constraints to ensure the proper behavior of the W and P predicates. First, three background constraints on the typing predicates P and W:

- $(\forall x)(Px \Rightarrow \sim Wx)$. [Propositions are not worlds.]
- $(\forall p)[Pp \Rightarrow (P(\neg p) \,\&\, P(\Box p) \,\&\, P(\Diamond p))]$. [If p is a proposition, then so are $\neg p$, $\Box p$ and $\Diamond p$.]
- $(\forall p)(\forall q)[(Pp \,\&\, Pq) \Rightarrow (P(p \wedge q) \,\&\, P(p \vee q) \,\&\, P(p \supset q) \,\&\, P(p \equiv q) \,\&\, P(p > q))]$. [If p and q are propositions, then so are $p \wedge q$, $p \vee q$, $p \supset q$, $p \equiv q$, and $p > q$.]

Next, our basic underlying semantical constraints on T and R (and all connectives, including classical ones):

- $(\forall p)(\forall w)[(Pp \,\&\, Ww) \Rightarrow ((T(\neg p)w \Leftrightarrow \sim Tpw) \,\&\, \sim(Tpw \,\&\, T(\neg p)w))]$.
- $(\forall p)(\forall q)(\forall w)[(Pp \,\&\, Pq \,\&\, Ww) \Rightarrow (T(p \wedge q)w \Leftrightarrow (Tpw \,\&\, Tqw))]$.
- $(\forall p)(\forall q)(\forall w)[(Pp \,\&\, Pq \,\&\, Ww) \Rightarrow (T(p \vee q)w \Leftrightarrow (Tpw \curlyvee Tqw))]$.
- $(\forall p)(\forall q)(\forall w)[(Pp \,\&\, Pq \,\&\, Ww) \Rightarrow (T(p \supset q)w \Leftrightarrow (Tpw \Rightarrow Tqw))]$.
- $(\forall p)(\forall q)(\forall w)[(Pp \,\&\, Pq \,\&\, Ww) \Rightarrow (T(p \equiv q)w \Leftrightarrow (Tpw \Leftrightarrow Tqw))]$.
- $(\forall p)(\forall w)[(Pp \,\&\, Ww) \Rightarrow (T(\Box p)w \Leftrightarrow (\forall w')(Ww' \Rightarrow Tpw'))]$. [Here, we're assuming S5 for $\Box$.]
- $(\forall p)(\forall w)[(Pp \,\&\, Ww) \Rightarrow (T(\Diamond p)w \Leftrightarrow (\exists w')(Ww' \,\&\, Tpw'))]$. [Here, we're assuming S5 for $\Diamond$.]
- $(\forall p)(\forall q)(\forall w)[(Pp \,\&\, Pq \,\&\, Ww) \Rightarrow (T(p > q)w \Leftrightarrow (\forall w')(Ww' \Rightarrow (Rpww' \Rightarrow Tqw')))]$.

The logic C is given by the above underlying definitions *alone*. With these basic underlying definitions in place, we are ready for constraints (1)–(7), which will be used to yield logics stronger than C:

1. $(\forall p)(\forall w)(\forall w')[(Pp \,\&\, Ww \,\&\, Ww') \Rightarrow (Rpww' \Rightarrow Tpw')]$.
2. $(\forall p)(\forall w)[(Pp \,\&\, Ww) \Rightarrow (Tpw \Rightarrow Rpww)]$.
3. $(\forall p)(\forall w)[(Pp \,\&\, Ww) \Rightarrow ((\exists w'')(Ww'' \,\&\, Tpw'') \Rightarrow (\exists w')(Ww' \,\&\, Rpww'))]$.
4. $(\forall p)(\forall q)(\forall w)(\forall w')[(Pp \,\&\, Pq \,\&\, Ww \,\&\, Ww') \Rightarrow (((Rpww' \Rightarrow Tqw') \,\&\, (Rqww' \Rightarrow Tpw')) \Rightarrow (Rpww' \Leftrightarrow Rqww'))]$.
5. $(\forall p)(\forall q)(\forall w)[(Pp \,\&\, Pq \,\&\, Ww) \Rightarrow ((\exists w')(Ww' \,\&\, Rpww' \,\&\, Tqw') \Rightarrow (\forall w'')(Ww'' \Rightarrow (R(p \wedge q)ww'' \Rightarrow Rpww'')))]$.
6. $(\forall p)(\forall w)(\forall w')(\forall w'')[(Pp \,\&\, Ww \,\&\, Ww' \,\&\, Ww'') \Rightarrow ((Rpww' \,\&\, Rpww'') \Rightarrow (w' = w''))]$.
7. $(\forall p)(\forall w)(\forall w')[(Pp \,\&\, Ww \,\&\, Ww') \Rightarrow ((Tpw \,\&\, Rpww') \Rightarrow (w = w'))]$.

The logic C^+ is given by C + (1)–(2). The logic S is given by C^+ + (3)–(5). The logic C_1 is given by S + (7). And, the logic C_2 is given by S + (6). Using this first-order formalization, we can give proofs in first-order logic of theorems and sequents of any of these logics, and we can also give counter-models. This is achieved *via* the following "translation scheme" from the language of conditional logics into first-order logic:

$$\Gamma \vDash_X p \text{ iff } \Gamma' \cup X' \vDash p'$$

where Γ is a set of statements in a conditional logic X, p is a statement of X, Γ' is the first-order translation of Γ, X' is the set of first-order constraints corresponding to the logic X, and p' is the first-order translation of p. In the next section, I will look at applications of this method to three Chapter 5 problems.

2 Three Illustrations of the Method

In this section, I will illustrate my first-order proof/counterexample method with applications to three examples from Chapter 5. Here, I will be using the natural deduction system for first-order logic, which is presented by Graeme Forbes in his introductory logic (12A) textbook *Modern Logic*.

2.1 $A > B \vDash_C A > (B \vee C)$

What we need to show is that the basic definitions *alone* entail the following (in first order logic):

$$(\forall w)[(Pa \,\&\, Pb \,\&\, Pc \,\&\, Ww) \Rightarrow (T(a > b)w \Rightarrow T(a > (b \vee c))w)].$$

Using our definitions, we can see that this reduces to proving the following *theorem* of FOL.[1]

$$(\forall x)[Wx \Rightarrow ((\forall y)((Pa \,\&\, Pb \,\&\, Wy) \Rightarrow (Raxy \Rightarrow Tby)) \Rightarrow (\forall z)((Pa \,\&\, Pb \,\&\, Pc \,\&\, Wz) \Rightarrow (Raxz \Rightarrow (Tbz \curlyvee Tcz)))].$$

Here is a Forbes-style natural deduction proof of this theorem of FOL:

1	(1)	Wd	Assumption (⇒I)
2	(2)	(∀y)(((Pa&Pb)&Wy)⇒(Rady⇒Tby))	Assumption (⇒I)
3	(3)	((Pa&Pb)&Pc)&We	Assumption (⇒I)
4	(4)	Rade	Assumption (⇒I)
2	(5)	((Pa&Pb)&We)⇒(Rade⇒Tbe)	2 ∀E
3	(6)	(Pa&Pb)&Pc	3 &E
3	(7)	Pa&Pb	6 &E
3	(8)	We	3 &E
3	(9)	(Pa&Pb)&We	7,8 &I
2,3	(10)	Rade⇒Tbe	5,9 ⇒E
2,3,4	(11)	Tbe	10,4 ⇒E
2,3,4	(12)	Tbe⋎Tce	11 ⋎I
2,3	(13)	Rade⇒(Tbe⋎Tce)	4,12 ⇒I
2	(14)	(((Pa&Pb)&Pc)&We)⇒(Rade⇒(Tbe⋎Tce))	3,13 ⇒I
2	(15)	(∀z)((((Pa&Pb)&Pc)&Wz)⇒(Radz⇒(Tbz⋎Tcz)))	14 ∀I
	(16)	(∀y)(((Pa&Pb)&Wy)⇒(Rady⇒Tby))⇒(∀z)((((Pa&Pb)&Pc)&Wz)⇒(Radz⇒(Tbz⋎Tcz)))	2,15 ⇒I
	(17)	Wd⇒((∀y)(((Pa&Pb)&Wy)⇒(Rady⇒Tby))⇒(∀z)((((Pa&Pb)&Pc)&Wz)⇒(Radz⇒(Tbz⋎Tcz))))	1,16 ⇒I
	(18)	(∀x)(Wx⇒((∀y)(((Pa&Pb)&Wy)⇒(Raxy⇒Tby))⇒(∀z)((((Pa&Pb)&Pc)&Wz)⇒(Raxz⇒(Tbz⋎Tcz)))))	17 ∀I

2.2 $\nvDash_{C_1} (A > B) \vee (A > \neg B)$

What we need to show is that the basic definitions + (1)–(5) + (7) do *not* entail the following (in FOL):

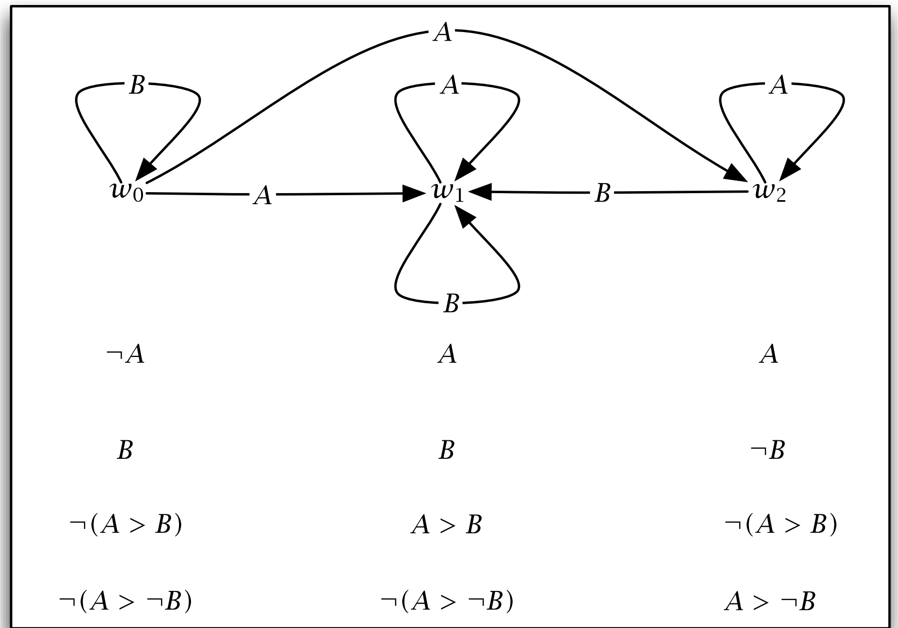
$$(\forall w)[(Pa \,\&\, Pb \,\&\, Ww) \Rightarrow T((a > b) \vee (a > \neg b))w].$$

[1] Note how the typing constraints on P and W aren't needed to prove this theorem. Typically, the typing constraints on P and W are not needed. But, when proving theorems (and especially finding models) in logics involving equality reasoning (*e.g.*, C_1 and C_2), one may need those additional constraints. While those constraints were not needed for either of the proofs reported on this handout, they were required for the generation of the (proper) C_1-counter-model reported in section **??**, below.

Using our definitions, we can see that this reduces to showing that the following claim does *not* follow from the basic definitions + (1)–(5) + (7) in FOL:

$$(\forall x)[Wx \Rightarrow ((\forall y)((Wy \,\&\, Pa \,\&\, Pb) \Rightarrow (Raxy \Rightarrow Tby)) \curlyvee (\forall z)((Wz \,\&\, Pa \,\&\, Pb) \Rightarrow (Raxz \Rightarrow {\sim}Tbz))].$$

Using a first-order model finder, I found the following counter-model.[2] It's just like the ones we've seen.



In this model, the world w_0 is the counterexample to the $\vDash_{C_1}$-claim in question, since both $\neg(A > B)$ and $\neg(A > \neg B)$ are true there, which makes $(A > B) \vee (A > \neg B)$ false there. And, as an exercise, you should make sure that constraints (1)–(5) + (7) are all satisfied in this model (hence making it a C_1-model).

2.3 $\vDash_{C_2} (A > B) \vee (A > \neg B)$

What we need to show is that the basic definitions + (1)–(6) *do* entail the following (in FOL):

$$(\forall w)[(Pa \,\&\, Pb \,\&\, Ww) \Rightarrow T((a > b) \vee (a > \neg b))w].$$

Using our definitions, we can see that this reduces to proving the following claim from (1)–(6) (in FOL).

$$(\forall x)[Wx \Rightarrow ((\forall y)((Pa \,\&\, Pb \,\&\, Wy) \Rightarrow (Raxy \Rightarrow Tby)) \Rightarrow (\forall z)((Pa \,\&\, Pb \,\&\, Pc \,\&\, Wz) \Rightarrow (Raxz \Rightarrow (Tbz \curlyvee Tcz)))].$$

In fact, this claim follows from (6) *alone*. Here is a Forbes-style natural deduction proof of this valid sequent:

[2] For those who are interested in playing around with theorem-provers and/or model finders on these sorts of problems, see me, and I'll give you an input file with all of the constraints, *etc.* Theorem provers and model-finders are able to prove all the theorems and generate all the counterexamples in the text. Indeed, they can solve much more difficult problems in these systems as well.

1	(1)	(∀x)(∀y)(∀z)(∀u)((((Px&Wy)&Wz)&Wu)⇒((Rxyz&Rxyu)⇒z=u))	Premise [(6)]
2	(2)	Wc	Ass (⇒I)
3	(3)	~((∀y)(((Pa&Pb)&Wy)⇒(Racy⇒Tby))⋎(∀z)(((Pa&Pb)&Wz)⇒(Racz⇒~Tbz)))	Ass (~I)
3	(4)	~(∀y)(((Pa&Pb)&Wy)⇒(Racy⇒Tby))&~(∀z)(((Pa&Pb)&Wz)⇒(Racz⇒~Tbz))	3 SI (Dem)
3	(5)	~(∀z)(((Pa&Pb)&Wz)⇒(Racz⇒~Tbz))	4 &E
3	(6)	(∃z)~(((Pa&Pb)&Wz)⇒(Racz⇒~Tbz))	5 SI (QS)
7	(7)	~(((Pa&Pb)&Wd)⇒(Racd⇒~Tbd))	Ass (∃E)
3	(8)	~(∀y)(((Pa&Pb)&Wy)⇒(Racy⇒Tby))	4 &E
3	(9)	(∃y)~(((Pa&Pb)&Wy)⇒(Racy⇒Tby))	8 SI (QS)
10	(10)	~(((Pa&Pb)&We)⇒(Race⇒Tbe))	Ass (∃E)
10	(11)	((Pa&Pb)&We)&~(Race⇒Tbe)	10 SI (Neg-Imp)
10	(12)	~(Race⇒Tbe)	11 &E
10	(13)	Race&~Tbe	12 SI (Neg-Imp)
10	(14)	~Tbe	13 &E
1	(15)	(∀y)(∀z)(∀u)((((Pa&Wy)&Wz)&Wu)⇒((Rayz&Rayu)⇒z=u))	1 ∀E
1	(16)	(∀z)(∀u)((((Pa&Wc)&Wz)&Wu)⇒((Racz&Racu)⇒z=u))	15 ∀E
1	(17)	(∀u)((((Pa&Wc)&Wd)&Wu)⇒((Racd&Racu)⇒d=u))	16 ∀E
1	(18)	(((Pa&Wc)&Wd)&We)⇒((Racd&Race)⇒d=e)	17 ∀E
10	(19)	(Pa&Pb)&We	11 &E
10	(20)	Pa&Pb	19 &E
10	(21)	Pa	20 &E
2,10	(22)	Pa&Wc	21,2 &I
7	(23)	((Pa&Pb)&Wd)&~(Racd⇒~Tbd)	7 SI (Neg-Imp)
7	(24)	(Pa&Pb)&Wd	23 &E
7	(25)	Wd	24 &E
2,7,10	(26)	(Pa&Wc)&Wd	22,25 &I
10	(27)	We	19 &E
2,7,10	(28)	((Pa&Wc)&Wd)&We	26,27 &I
1,2,7,10	(29)	(Racd&Race)⇒d=e	18,28 ⇒E
7	(30)	~(Racd⇒~Tbd)	23 &E
7	(31)	Racd&~~Tbd	30 SI (Neg-Imp)
7	(32)	Racd	31 &E
10	(33)	Race	13 &E
7,10	(34)	Racd&Race	32,33 &I
1,2,7,10	(35)	d=e	29,34 ⇒E
7	(36)	~~Tbd	31 &E
1,2,7,10	(37)	~~Tbe	35,36 =E
1,2,7,10	(38)	Tbe	37 DN
1,2,7,10	(39)	Λ	14,38 ~E
1,2,3,7	(40)	Λ	9,10,39 ∃E
1,2,3	(41)	Λ	6,7,40 ∃E
1,2	(42)	~~((∀y)(((Pa&Pb)&Wy)⇒(Racy⇒Tby))⋎(∀z)(((Pa&Pb)&Wz)⇒(Racz⇒~Tbz)))	3,41 ~I
1,2	(43)	(∀y)(((Pa&Pb)&Wy)⇒(Racy⇒Tby))⋎(∀z)(((Pa&Pb)&Wz)⇒(Racz⇒~Tbz))	42 DN
1	(44)	Wc⇒((∀y)(((Pa&Pb)&Wy)⇒(Racy⇒Tby))⋎(∀z)(((Pa&Pb)&Wz)⇒(Racz⇒~Tbz)))	2,43 ⇒I
1	(45)	(∀x)(Wx⇒((∀y)(((Pa&Pb)&Wy)⇒(Raxy⇒Tby))⋎(∀z)(((Pa&Pb)&Wz)⇒(Raxz⇒~Tbz))))	44 ∀I