Tableau-Based Theorem Proving in Normal Conditional Logics

Chris Groeneboer
James P. Delgrande

School of Computing Science,
Simon Fraser University,
Burnaby, B.C.,
Canada V5A 1S6

ABSTRACT

This paper presents an extension of the semantic tableaux approach to theorem proving for the class of normal conditional logics. These logics are based on a possible worlds semantics, but contain a binary "variable conditional" operator $\Rightarrow$ instead of the usual operator for necessity. The truth of $A \Rightarrow B$ depends both on the accessibility relation between worlds, and on the proposition expressed by the antecedent A. Such logics have been shown to be appropriate for representing a wide variety of commonsense assertions, including default and prototypical properties, counterfactuals, notions of obligation, and others.

The approach consists in attempting to find a truth assignment which will falsify a sentence or set of sentences. If successful, then a specific falsifying truth assignment is obtained; if not, then the sentence is valid. The approach is arguably more natural and intuitive than those based on proof-theoretic methods. The approach has been proven correct with respect to determining validity in the class of normal conditional logics. In addition, the approach has been implemented and tested on a number of different conditional logics. Various heuristics have been incorporated, and the implementation, while exponential in the worst case, is shown to be reasonably efficient for a large set of test cases.

1. Introduction

It is by now generally accepted in Artificial Intelligence (AI) that statements of default or prototypical properties cannot easily or obviously be represented in classical logic by means of the material conditional. For example, statements such as "birds fly", along with "penguins are birds", and "penguins do not fly" have no ready consistent translation into classical logic, unless one is willing to say that there are no penguins. Similarly, "ravens are black" and "albino ravens are not black" have no ready consistent translation, unless there are no albino ravens. In the first case, transitivity of the conditional is denied and, in the second, a strengthening of the antecedent results in a denial of the original consequent. Approaches in AI to address such statements include providing a schema for adding formulae to a set of sentences [McCarthy 80], [McCarthy 84], and extending classical first-order logic by the addition of rules of inference [Reiter 80], or the addition of unary or binary sentential operators [McDermott and Doyle 80; Delgrande 87].

What is perhaps less well-known in Artificial Intelligence is that such non-standard patterns occur in other types of reasoning. For example, it seems quite reasonable to make the following counterfactual assertions [Lewis 73]:

> "If John had gone to the party, it would have been a good party" along with
> "If John and Sue had gone, it would not have been a good party" but
> "If John, Sue, and Mary had gone, it would have been a good party".

or the following hypothetical assertions [Nute 80]:

> "If John were to work less, he would be less tense",
> "If John were to lose his job, he would work less", together with
> "If John were to lose his job, he would not be less tense".

Assertions of obligation also conform to similar patterns. Thus, for example, one should safely prevent a crime from occurring, if possible, unless preventing that crime would cause a greater one. These examples all seem to be reasonable, consistent, commonsense assertions to make about the external world, and hence reasonable statements to represent and reason about. However these examples have no straightforward translation into classical logic. In philosophy, the general framework of *conditional logics* has been used for formalising such reasoning. The general idea is that an operator, $\Rightarrow$, called the *variable conditional*, is introduced into either classical propositional or first-order logic. The truth of a statement $A \Rightarrow B$ however relies not on the present state of affairs being modelled, but on other, alternative, states of affairs (or *possible worlds*). In addition there is a binary *accessibility* relation among these states of affairs, where the relation provides an ordering among worlds. The truth of $A \Rightarrow B$ then is determined by considering the "least" worlds in which A is true; if B is also true in all such worlds, then $A \Rightarrow B$ is true.

This paper presents an extension to the method of *semantic tableaux* [Smullyan 68; Hughes and Cresswell 68] for determining the validity of a set of sentences in a wide class of conditional logics, called *normal conditional logics*. The general idea is that for a sentence ω, an attempt is made to construct a model for $\neg\omega$. If such a model is obtained, ω is not valid, and moreover one has a specific falsifying interpretation for ω. Otherwise such a model is shown to be impossible to construct, and ω is valid. The class of logics to which this approach may be applied includes those for subjunctive conditionals (including counterfactual and hypothetical conditionals), default conditionals, and deontic conditionals. The general approach, being based on the model theory of such systems, is arguably more intuitive than resolution-based or other proof-theory based methods.

The next section introduces and discusses conditional logics in a bit more detail. Section 3 introduces theorem proving based on semantic tableaux in general, along with the present approach; section 4 discusses the approach in detail. The fifth section describes an implementation of the method, while the last section provides concluding remarks. This work derives from that presented in [Groeneboer 87], which developed a theorem-prover for conditional statements in a logic of defaults; further details, proofs of theorems, etc. may be found therein.

2. Conditional Logics

Consider again the example concerning the potential, but counterfactual, outcomes of a past party. We might write the sentences in a propositional system as:

$$John\_went \Rightarrow Good\_party$$
$$John\_went \wedge Sue\_went \Rightarrow \neg Good\_party$$
$$John\_went \wedge Sue\_went \wedge Mary\_went \Rightarrow Good\_party.$$

The pattern in these statements is clear: as the antecedent is strengthened, the consequent may change to its negation. In classical logics of course a set of such conditionals is satisfiable only if some of the antecedents are false. In a similar fashion, the laws of transitivity of the conditional, and of the contrapositive, may be shown to be violated for counterfactual assertions.

Such patterns of inference and deviations from the classical norm have been recognised by philosophers in a number of types of reasoning. Best known in this regard is counterfactual reasoning [Stalnaker 68; Lewis 73]. A counterfactual statement is one in which, among other things, the antecedent is false in the state of affairs being modelled, but the conditional as a whole may be either true or false. In Lewis's approach, the counterfactual $A \Rightarrow B$ is true if in the set of worlds most similar to our own where A is true, B is true also. Thus, the counterfactual "if John had come it would have been a good party" is true if, in the worlds closest (or, most similar) to our own in which John did in fact come to the party, it was a good party. More formally, the truth of counterfactuals is determined using a possible worlds semantics, where the accessibility relation between worlds corresponds to a notion of "similarity". A counterfactual $A \Rightarrow B$ is true in model M at a particular world w just when the closest (according to the accessibility relation) worlds in which A is true also have B true. Appropriate properties of the conditional then are obtained by imposing suitable restrictions on the accessibility relation. Counterfactual reasoning is in turn an example of subjunctive reasoning; [Pollock 76] identifies and addresses four broad categories of such reasoning. In addition, deontic logics of conditional obligation [van Fraassen 72; Chellas 80] also allow similar patterns of satisfiability. In AI it has more recently been demonstrated that default and prototypical properties may also be expressed within such a framework [Glymour and Thomason 84; Delgrande 87]. A formal description of counterfactuals which allows applications in areas such as diagnosis is presented in [Ginsberg 85]. See [Chellas 75] and [Nute 80] also for general discussions of approaches to such reasoning.

What we have then with conditional logics is a class of modal logics in which the modal operator is binary, rather than unary, as is usually the case for notions of necessity, knowledge, time, etc. The various logics differ then depending on the conditions imposed on the accessibility relation, and on how the set of worlds selected depends on the antecedent of the conditional. As we impose different restrictions on this accessibility relation, we obtain different conditional logics with differing characteristics. Thus for example, most counterfactual logics contain $(A \wedge (A \Rightarrow B)) \supset B$ as a theorem – that is, if the antecedent of the (supposed) counterfactual is true, then so is the consequent. For a logic for defaults however [Delgrande 87], one is typically interested in the situation where the antecedent of the conditional is true but the consequent may not be. Thus one may want to assert that "ravens are normally black", while allowing for non-black ravens in the present state of affairs.

However, all the logics we will deal with, the *normal conditional logics*, contain the following rules of inference:

RCK If $(B_1 \wedge \cdots \wedge B_n) \supset B$ then $((A \Rightarrow B_1) \wedge \cdots \wedge (A \Rightarrow B_n)) \supset (A \Rightarrow B)$

RCEA If $A \equiv A'$ then $(A \Rightarrow B) \equiv (A' \Rightarrow B)$.

These relations can be compared with the rule of inference characterising the *normal modal logics* of necessity [Chellas 80], where the formula LB is read as "B is necessarily true":

RK If $(B_1 \wedge \cdots \wedge B_n) \supset B$ then $(LB_1 \wedge \cdots \wedge LB_n) \supset LB$.

The minimal normal logic of necessity is the system K.

3. Semantic Tableaux

The approach to theorem proving for conditional logics presented in this paper is a tableau-based method. The roots of such systems can be traced back to [Gentzen 69]. [Smullyan 68] applied these ideas to classical logic, simplifying the methods and making them more elegant. Tableau systems have also evolved for modal logics [Kripke 63; Hughes and Cresswell 68; Fitting 83] and temporal logics [Rescher and Urquhart 71].

The tableau method involves attempting to construct a model for $\neg\omega$ in order to prove a formula ω. If a model is found for $\neg\omega$ then ω cannot be valid; otherwise $\neg\omega$ is unsatisfiable and ω is valid. For classical propositional logic the method is straightforward. Consider for example the formula $(A \wedge B) \supset (A \vee B)$. The goal is to construct a falsifying interpretation for the formula. Because the main connective is a material conditional, for the formula to be false, the antecedent must be true and the consequent false; for the antecedent to be true, both A and B must be true. However, this requires that the consequent be true. Hence a falsifying interpretation cannot be found, and the original formula is valid.

This approach is usually illustrated by initially placing a "0" (for falsity) under the main connective, and then successively placing the values "0" or "1" under the other connectives. There are two types of rules for specifying how values are to be assigned, called α-rules and β-rules. For α-rules, there is no choice as to how values may be assigned to a subexpression, given a value for the main connective. Thus, if the formula $A \vee B$ has a 0 under the main connective, then the only way the formula can be falsified is if A and B are false. For the other case, β-rules, alternatives may be generated. Thus, to falsify $A \wedge B$, one need falsify either A or B. In the case of β-rules, the formula is replicated to allow for the various alternatives, and a successful assignment of values to any of the alternatives succeeds in satisfying the original formula. Since β-rules spawn a number of alternatives, clearly it is preferable to apply α-rules first wherever possible.

The approach generalises elegantly to modal logics. Again we attempt to construct a falsifying model for a sentence. However the modal operators L for *necessity* (or, truth in all accessible worlds) and M for *possibility* (or truth in some possible world) require further machinery. Thus, the formula MA requires that there be a world in which A is true; in this case we create a new world in which A is true, and indicate that that world is accessible from the first. If however we were attempting to falsify MA, we would require that A be false in all worlds accessible from the first. Hence again we have α-rules and β-rules (which generate a single alternative or multiple alternatives, respectively) for the modal operators. See [Hughes and Cresswell 68] for details.

Consider then the case of a general conditional logic. The language of the logic is that of propositional logic augmented with a binary conditional operator $\Rightarrow$. Truth of a sentence in the logic is determined with respect to a model structure $M = <W, E, P>$ where informally W is a set of possible worlds, E is a binary accessibility relation between possible worlds, and P is a mapping of primitive propositions onto possible worlds. (Thus P determines which primitive propositions are true at which worlds.) The truth of the standard connectives at a possible world is determined by the usual recursive definition; for example $A \vee B$ is true at a world $w \in W$ just when either A or B is true at w. Hence the method of semantic tableaux can be applied to formulae composed from the classical connectives, except that now sentences are indexed by worlds.

For truth of the variable conditional at a world, there are two possibilities. First, $A \Rightarrow B$ is true at world w just when the "closest" worlds to w in which A is true also have B true. In other words, $A \Rightarrow B$ is true just when there is a world w_1 in which both A and B are true, and for any w_2 accessible from w_1 it must be the case that $A \supset B$ is true. The second possibility takes care of the situation in which there are no accessible worlds where A is true. In this case $A \Rightarrow B$ is taken as being vacuously true. For the falsity of the variable conditional, we have that there is some world w_1 in which A is true and B false, and if w_2 is a world accessible from w_1 with both A and B true, then w_1 is also accessible from w_2.

What this means is that these considerations impose a set of constraints on the structure of worlds in a model, regardless of the conditional logic involved. So the first step is to generate a structure, called the *semi-complete structure*, which specifies initial constraints that are required for any model. The semi-complete structure then implicitly constrains the class of models for the sentence in question. From this structure, individual *templates* of models are generated; these templates consist of a set of worlds, together with a minimal set of accessibility relations between worlds. Thirdly, for each such template, conditions on the accessibility relation (for example, transitivity, reflexivity, etc.) are imposed; these conditions will vary from logic to logic. If for any of the augmented templates a model of the original sentence $\neg\omega$ is obtained, then the sentence $\neg\omega$ has been satisfied and hence, a specific falsifying interpretation for ω has been found. If no such model is found then ω cannot be falsified and is valid. The use of semi-complete structures and templates extends the procedure given in [Hughes and Cresswell 68]. The approach provably provides a decision procedure for the class of normal conditional logics. In addition, the algorithmic nature of the approach leads to a straightforward and intuitive implementation. The next section describes this approach in more detail.

4. A Theorem Prover for Normal Conditional Logics

The steps taken in attempting to construct a falsifying interpretation will be discussed in turn. However, first we describe the graphical notation used for forming semi-complete structures. Truth conditions for the $\Rightarrow$ operator are given in terms of the diagrams of Figure 1.

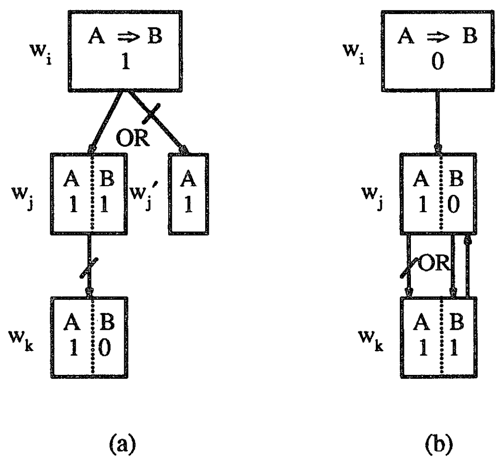

Figure 1.

For Figure 1(a), we wish to construct a structure for which $A \Rightarrow B$ is true at world w_i; this is represented by the topmost box. This conditional is true when one of two conditions occurs, and is indicated by the diverging arrows labelled jointly with an "OR". The left arrow specifies that there is some accessible world w_j, in which A and B are individually true, and there is no world w_k accessible from w_j (given by the arrow with the slash) in which A is true and B false. The right arrow from w_i specifies that for no accessible world is A true, thus A is necessarily false in this alternative. For Figure 1(b), $A \Rightarrow B$ is false at a world if there is some accessible world w_j wherein A is true and B false, and from this world one of two alternatives obtain: either there is no accessible world in which A and B are both true, or else if such a world exists, then w_j is accessible from it.

We use the following notation then for the construction of a model:

1. rectangles, possibly labelled, represent worlds,
2. sets of formulae with "forced" truth values within a rectangle w_i indicate what must be true or false at w_i,

3. arrows represent accessibility between worlds,
4. not-arrows, $\not\rightarrow$, which emanate from labelled rectangles, w_i, and enter other rectangles. These arrows denote constraints on worlds accessible from w_i,
5. OR's denote alternative states of affairs.

Three types of rules are used for the assignment of truth values: α-rules, β-rules, and γ-rules. α-rules and β-rules are applied as in the propositional case to the classical connectives. The γ-rules capture the truth conditions for the variable conditional:

1. $\models_w^M A \Rightarrow B$ iff (a) there exists a w_1 such that Eww_1 and $\models_{w_1}^M A$ and $\models_{w_1}^M B$ and there exists no w_2 such that Ew_1w_2 and $\models_{w_2}^M A$ and $\models_{w_2}^M \neg B$, or (b) for all w_1 such that Eww_1, $\models_{w_1}^M \neg A$.
2. $\models_w^M \neg(A \Rightarrow B)$ iff there is a w_1 where Eww_1 and $\models_{w_1}^M A$ and $\models_{w_1}^M \neg B$ and either (a) there is no w_2 such that Ew_1w_2 and $\models_{w_2}^M A$ and $\models_{w_2}^M B$, or (b) if there is such a w_2 then Ew_2w_1.

The diagrams of Figure 1 graphically express these rules. Thus:

1. If $A \Rightarrow B$ is assigned 1 at w_i, create a new rectangle w_j in which the antecedent is true and the consequent is true (Figure 1(a)). Place an arrow from w_i to new rectangle w_j. Create another rectangle w_j' in which the antecedent is true. Place a not-arrow from w_i to w_j'. Place an OR between the arrow from w_i to w_j and the not-arrow from w_i to w_j'. Create another rectangle w_k in which the antecedent is true and the consequent false. Place a not-arrow from w_j to w_k.
2. If $A \Rightarrow B$ is assigned 0 at w_i, create a new rectangle w_j in which the antecedent is true and the consequent false (Figure 1(b)). Place an arrow from w_i to w_j. Create another rectangle w_k in which the antecedent and the consequent are both true. Place a not-arrow from w_j to w_k. Place an arrow from w_j to w_k and from w_k to w_j. Place an OR between the not-arrow and the arrow connecting w_j and w_k.

There are three steps in attempting to provide a falsifying interpretation for a sentence ω:

1. Build the semi-complete structure(s)
2. Generate templates
3. Repeat until all configurations are tested or a model is found:
   - 3.1. Determine a set of accessibility constraints
   - 3.2. Test obtained configuration

Note that Step 1 does not yield a model (because it contains "OR's" and arrows with slashes through them) but rather yields a "proto-model" from which models may be generated. Step 2 is concerned with generating a model in the basic conditional logic – that is, the logic in which there are no constraints placed on the accessibility relation. Step 3.1 is logic-specific, and is concerned with enforcing the constraints imposed by a particular accessibility relation. The structure obtained after accessibility constraints are enforced is referred to as a *configuration*. In Step 3.2 each configuration is tested to determine whether or not it is indeed a model for the original formula. Note that it is a trivial modification to extend the system to a first-order theorem prover. One need only replace the application of α-rules and β-rules at a world with a first-order component [Smullyan 68].

In this and the next section we will use the formula $\omega = ((A \Rightarrow C) \wedge (B \Rightarrow C)) \supset (A \Rightarrow (B \wedge C))$ and the system of [Delgrande 87] to illustrate the method. This system, which is intended for representing default properties, is a normal conditional logic in which the accessibility relation is reflexive, transitive and forward connected (that is, if Ew_1w_2 and Ew_1w_3 then either Ew_2w_3 or Ew_3w_2). We begin by writing ω in a rectangle labelled w_1 and placing a 0 under the main operator. Then α-rules, β-rules, and γ-rules are applied as often as possible to obtain the semi-complete structure of Figure 2. Those formulae which are substitution instances of tautologies in standard propositional logic are determined to be valid at this point.

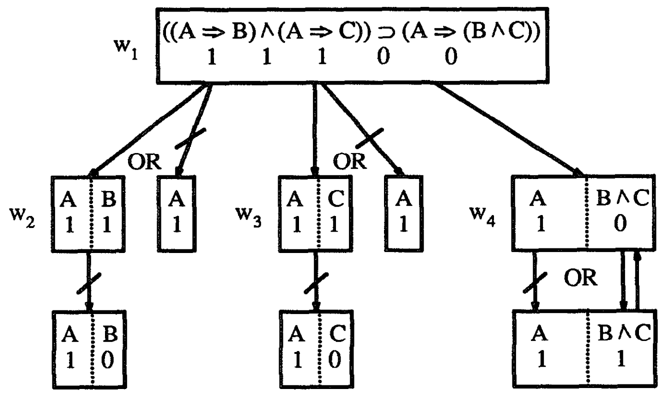

Figure 2. Example semi-complete structure.

Templates are generated from the semi-complete structure in Step 2. This template-generation step yields each of the possibilities afforded by the combination of OR's. Referring to Figure 2, note that templates in which A is necessarily false cannot yield a model because A must be true at w_4. So in this case we continue with just one template, that of Figure 3. The validity of formulae valid in all normal conditional logics is determined at this point – for example, $A \Rightarrow (A \vee \neg A)$.

In Step 3.1 accessibility constraints are enforced. Recall that the accessibility relation of our example is reflexive, transitive, and forward connected. This means that sets of accessibility arrows must be added to the template which connect the worlds in the template in such a way that the properties comprising the accessibility relation hold. In the example there are 13 distinct ways in which to enforce accessibility constraints. Since no new worlds are added, the process is guaranteed to terminate.

For the remainder of the paper we will use the notation Aw_iw_j to indicate that there is a "candidate" accessibility between worlds w_i and w_j. Step 3.2 then involves testing each candidate set of possible accessibility relation instances to determine whether or not it is a model. For example, the configuration obtained when the set Aw_2w_3, Aw_2w_4, Aw_3w_4

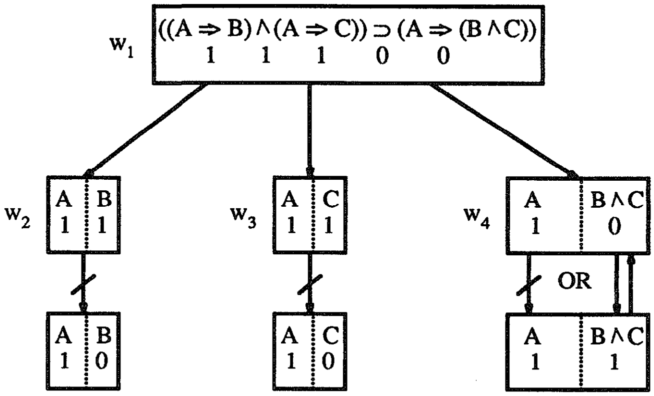

Figure 3. Example structural template.

is added to the template does not form a model. The reason for this is that because of "forced" values at worlds, Aw_2w_4 and Aw_3w_4 cannot coexist. The constraints on accessibility from w_2 require that B be true at w_4, whereas the constraints on accessibility from w_3 require that C be true at w_4. But $B \wedge C$ is false at w_4. None of the 13 configurations provides a model for $\neg\omega$, and ω is therefore valid. Note that the same formula is invalid in a logic in which the accessibility relation is only reflexive. The template of Figure 3 (minus the accessibility constraints) serves as a counterexample.

5. Implementation Considerations

A semi-complete structure is represented as a table with three columns: (1) world labels, (2) formulae true at a world, and (3) conditions on accessible worlds. Table 1 represents the semi-complete structure of Figure 2. Thus for example, consider the accessibility constraints column for world w_4. The (not particularly elegant) notation indicates that for any world w_i accessible from w_4, it must be the case that $\neg A \vee \neg(B \wedge C)$ is true at w_i, or else w_4 is accessible from w_i. It is from this table that templates are generated.

Table 1. Semi-complete structure for ω

worlds	conditions at world	accessibility constraints
w_1	$\neg\omega$	$(Aw_1w_2 \vee \neg A) \wedge (Aw_1w_3 \vee \neg A) \wedge Aw_1w_4$
w_2	$A \wedge B$	$\neg A \vee B$
w_3	$A \wedge C$	$\neg A \vee C$
w_4	$A \wedge \neg(B \wedge C)$	$(\neg A \vee \neg(B \wedge C)) \vee Aw_iw_4$

A number of heuristics and other techniques are employed to improve efficiency. A simple, but inefficient approach to arrow-set generation is to generate all possible arrow sets which connect the template, then test each for the properties imposed by the accessibility relation. Alternatively, we can use logic-specific heuristics to constrain this generation. Thus for example the fact that equivalence classes of worlds form an integral part of the semantics in the logic described in [Delgrande 87] is exploited in [Groeneboer 87] to provide a more efficient generation component for a theorem prover for that system.

Another technique is used to improve efficiency of the testing phase. Each potential accessibility Aw_iw_j is examined once to determine what subformulae must be true at w_j so that constraints on accessibility from w_i are not violated. The results are stored in a table called the table of forced values. For example, Table 2 gives the forced values for ω. Consider potential accessibility Aw_2w_3. A and C are both true at w_3, thus B must also be true at w_3 so that Aw_2w_3 does not violate constraints on accessibility from w_2.

Table 2. Forced values for ω

Arrows	A	B	C	B∧C
Aw_3w_2	1	1	1	
Aw_4w_2	1	1	0	
Aw_2w_3	1	1	1	
Aw_4w_3	1	0	1	
Aw_2w_4	1	1	0	0
Aw_3w_4	1	0	1	0

In building the semi-complete structure, α-rules are applied whenever possible before β-rules. For the example, we left w_4 as is, avoiding creation of three alternative semi-complete structures, each differing only in the assignment of truth values to B and C at w_4. The fact that $B \wedge C$ is false at w_4 is retained in Table 2.

A further efficiency-improvement technique involves the use of another table which gives the consequences of the information in the table of forced values. The table of accessibility constraints derived from Table 2 is given in Table 3. Aw_2w_4 requires that B be true at w_4, whereas Aw_3w_4 requires that B be false at w_4. Therefore any configuration in which both accessibilities occur cannot be a model. This is made explicit in constraint (1) of Table 3.

Table 3. Arrow constraints for ω

(1) Aw_2w_4 and Aw_3w_4 cannot coexist
(2) If Aw_3w_2 exists then Aw_2w_4 and Aw_3w_4 must coexist
(3) If Aw_2w_3 exists then Aw_3w_4 and Aw_2w_4 must coexist

It might at first appear that Aw_3w_2 and Aw_4w_2 are similarly incompatible, but recall that the accessibility constraints from w_4 contain an OR. So if Aw_3w_2 and Aw_4w_2 occur then Aw_2w_4 must also occur. If Aw_3w_2 occurs and Aw_4w_2 does not, then Aw_2w_4 must occur by the forward-connectedness property. Thus if Aw_3w_2 occurs then Aw_2w_4 must co-occur, whether or not Aw_4w_2 is present. If Aw_3w_2 and Aw_2w_4 coexist, then Aw_3w_4 must exist by transitivity. This is made explicit in constraint (2) of Table 3. Constraint (3) is derived similarly from Table 2.

Observe that we can derive from Table 3 the fact that no model of $\neg\omega$ is possible since (a) Aw_3w_2 cannot occur by constraints (1) and (2), (b) Aw_2w_3 cannot occur by constraints (1) and (3), but (c) one of Aw_2w_3, Aw_3w_2 must occur by forward-connectedness. But this is a contradiction. In the example then no reflexive, transitive, forward connected set of accessibilities is possible which would provide a model for $\neg\omega$, and ω is therefore valid.

With regard to complexity considerations, the procedure is clearly exponential in the worst case. It has however been tested in a number of differing logics and on a set of about 25 test formulae of varying complexity. In nearly all cases the procedure appeared roughly linear in the size of a formula.

The complexity of each of the three steps in attempting to find a falsifying interpretation are easily determined. Building the semi-complete structure is clearly linear in the number of occurrences of the connective $\Rightarrow$ in a formula. Since the semi-complete structure can have n OR's, where n is the number of occurrences of $\Rightarrow$, template generation is potentially exponential. Similarly, for each template, generating a configuration is potentially exponential, since one is effectively testing various possible orderings of worlds.

6. Conclusion

This paper has presented an extension of the semantic tableaux approach to theorem proving for a wide class of conditional logics; such logics contain a "variable conditional" $\Rightarrow$, where $A \Rightarrow B$ is true if the "closest" ("simplest", or whatever) worlds in which A is true have B true also. Different logics are obtained as the notion of accessibility between worlds is altered. These logics are appropriate for representing a wide and useful set of types of commonsense assertions, and have been used not only for representing default and prototypical properties, but also counterfactuals and hypotheticals, notions of obligation, etc.

The approach consists in attempting to find a truth assignment which will falsify a sentence, or set of sentences. If successful, then a specific falsifying assignment is obtained; if not, then the sentence is valid. Since the method is based on the model theory of the systems involved, it provides an arguably more natural and intuitive approach than others based on the proof-theories of the systems. The approach has been proven to exactly capture validity for this class of logics.

The approach has been implemented (in Franz Lisp) and tested on a number of different logics. Various heuristics have been incorporated in the implementation and, while the algorithm is exponential in the worst case, it is reasonably efficient for a large set of test cases. The implementation breaks the problem into a natural sequence of steps in the attempt to find a falsifying assignment. Hence heuristics specific to a particular part of the problem may be easily incorporated. Since accessibility relation restrictions are the last to be enforced, the program is easily modified to deal with different logics.

However, only the propositional case has been implemented. The reason for this is that we are concerned only with that aspect of theorem proving dealing with the conditional operator. Since the Barcan formula and its inverse would be valid in the first-order analogue of the logics that we consider, first-order reasoning could be "localised" at worlds, and would not interact with the conditional operator; hence the first-order case could be trivially incorporated in the prover.

Acknowledgements

This research was supported in part by the National Science and Engineering Research Council of Canada grant A0884 and in part by a grant from Simon Fraser University.

References

B.F. Chellas, "Basic Conditional Logic", Journal of Philosophical Logic 4, 1975, pp 133-153.

B.F. Chellas, *Modal Logic: An Introduction*, Cambridge University Press, 1980.

J.P. Delgrande, "A First-Order Logic for Prototypical Properties", *Artificial Intelligence 33, 1*, 1987, pp 105-130.

M.C. Fitting, *Proof Methods for Modal and Intuitionistic Logics*, Reidel Publishing, Dordrecht, Holland, 1983.

G. Gentzen, "Investigations into Logical Deduction", in *The Collected Papers of Gerhard Gentzen*, M.E. Szabo, (ed.), North-Holland Publishing Company, Amsterdam, 1969, pp 68-131.

M.L. Ginsberg, "Counterfactuals", *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 1985, pp 80-86.

C. Glymour and R.H. Thomason, "Default Reasoning and the Logic of Theory Perturbation", *Proceedings of the Non-Monotonic Reasoning Workshop*, American Association for Artificial Intelligence, 1984, pp 93-102.

R.C. Groeneboer, "Tableau-Based theorem Proving for a Conditional Logic", M.Sc. thesis, School of Computing Science, Simon Fraser University, 1987.

G.E. Hughes and M.J. Cresswell, *An Introduction to Modal Logic*, Methuen and Col. Ltd., 1968.

S. Kripke, "Semantical Considerations on Modal Logics", *Acta Philosophica Fennica, Modal and Many-valued Logics*, 1963, pp 83-94.

D. Lewis, *Counterfactuals*, Harvard University Press, 1973.

J. McCarthy, "Circumscription – A Form of Non-Monotonic Reasoning", *Artificial Intelligence 13*, 1980, pp 27-39.

J. McCarthy, "Applications of Circumscription to Formalizing Commonsense Reasoning", *Proceedings of the Non-Monotonic Reasoning Workshop*, American Association for Artificial Intelligence, 1984, pp 295-324.

D. McDermott and J. Doyle, "Non-Monotonic Logic I", *Artificial Intelligence 13*, 1980, pp 41-72.

E. Mendelson, *Introduction to Mathematical Logic*, D. Van Nostrand Co., 1964.

D. Nute, *Topics in Conditional Logic*, Philosophical Studies Series in Philosophy, Volume 20, D. Reidel Pub. Co., 1980.

J. Pollock, *Subjunctive Reasoning*, Philosophical Studies Series in Philosophy, Reidel Publishing, Dordrecht, Holland, 1976.

R. Reiter, "A Logic for Default Reasoning", *Artificial Intelligence 13*, 1980, pp 81-132.

N. Rescher and A. Urquhart, *Temporal Logic*, Springer-Verlag/Wien, 1971.

R.M. Smullyan, *First-Order Logic*, Springer-Verlag, 1968.

R.F. Stalnaker, "A Theory of Conditionals", in *Studies in Logical Theory*, N. Rescher (ed.), Basil Blackwell, Oxford, 1968, pp 98-112.

B.C. van Fraassen, "The Logic of Conditional Obligation", *Journal of Philosophical Logic 1*, 1972, pp 417-438.