Connexive Semantic Tableaux

Storrs McCall

Abstract

Connexive logic differs from other alternatives to classical two-valued logic in containing theses such as ~(p→~p) and (p→q)→~(p→~q) which are two-valuedly false. In this paper a semantic tableau formulation of connexive logic is constructed, using classical tableau rules plus an additional rule enabling a contradiction to be derived from any tableau item of the form A→~A or ~A→A. The resulting system is shown to be sound in the sense of never leading to contradiction, and complete in the sense that all valid connexive theses are derivable.

Introduction

Connexive implication takes its inspiration from remarks about conditionals A→B made by Sextus Empiricus in the 4^{th} century BC, when it was said that “the very crows on the roof-tops were croaking about what conditionals were true”. Sextus held that A→B was true when ~B, the contradictory of B, was incompatible with A, and this would imply that no conditional of the form p→~p could be true, since the contradictory of ~p is never incompatible with p (see McCall (2012) and (2014) for historical details. Wansing (2014) is an excellent general introduction.)

Connexive systems are unlike all other non-classical logics in not being sub-systems of two-valued logic. The connexive formula ~(p→~p) takes the value “F” in two-valued logic when p takes the value “F”. In 1962 R.B. Angell produced four-valued matrices which satisfy ~(p→~p) and (p→q) → ~(p→~q), in addition to many two-valued tautologies. These matrices are axiomatized in McCall (1966). McCall (2014) contains a Gentzen *Sequenzen-kalkül* formulation of connexive logic, and the aim of the present paper is to construct a simple semantic tableau system for connexive theses.

Tableau rules for classical propositional logic are found in many introductory texts, e.g. Jeffrey (1967):

A→B	~(A→B)	A&B	~(A&B)	A∨B	~(A∨B)	~~A
↙ ↘	A	A	↙ ↘	↙ ↘	~A	A
~A B	~B	B	~A ~B	A B	~B	

Filosofiska Notiser, Årgång 2, Nr. 2, September 2015, 3-18

The following tableau illustrates the use of these rules. Suppose we wish to show that Syl, the formula (p→q) → [(q→r)→(p→r)], is valid. We assume that Syl is false, and try to produce contradictions in every branch of the tableau:

1. ~{(p→q) → [(q→r)→(p→r)]}
2. p→q [1]
3. ~[(q→r)→(p→r)] [1]
4. q→r [3]
5. ~(p→r) [3]
6. p [5]
7. ~r [5]
↙ ↘
8. ~p   q [2]
x   ↙ ↘
9. ~q   r [4]
x   x

An “x” at the bottom of a branch indicates that the branch in question is closed. All three branches in the tableau where Syl is assumed to be false are closed, and consequently Syl is valid.

Tableau rules for non-classical connexive theses such as ~(p→~p), known as “Aristotle”, and (p→q) → ~(p→~q), “Boethius”, require an additional rule, which we may name “Aristotle”. The rule Aristotle takes the following two forms:

A→~A	~A→A
A	~A
~A	A

Here is the tableau for the thesis Aristotle, where the rule DN permits the derivation of a wff A from ~~A at any point in a tableau:

1. ~~(p→~p)
2. p→~p [1, double negation (DN)]
3. p [2, the rule Aristotle]
4. ~p [2]
5. x

In the tableau for Boethius we shorten the work by making use of two new derived rules, the rule “Syl” and the rule “Contra”. The rule “Syl” states that if we have a line in a branch of the form "A→B", and another line in the same branch of the form "B→C", then we may add a line "A→C". The rule "Contra" (from the word "contraposition") has 4 variants:

(1) If we have a line of the form A→B, we can add a line of the form ~B→~A.

The tableau derivation of the thesis (A→B) → (~B→~A) corresponding to (1) is as follows:

1. ~[(A→B) → (~B→~A)]
2. A→B [1]
3. ~(~B→~A) [1]
4. ~B [3]
5. ~~A [3]

↙ ↘

6. ~A B [2]

x x

(2) From A→~B derive B→~A.
(3) From ~A→B derive ~B→A.
(4) From ~A→~B derive B→A.

The tableaux corresponding to the rules Contra (2), (3) and (4) are similar to that for (1).

Here is Boethius:

1. ~[(p→q) → ~(p→~q)]
2. p→q [1]
3. ~~(p→~q) [1]
4. p→~q [3, DN]
5. q→~p [4, Contra (2)]
6. p→~p [2, 5 Syl]
7. p [6, Aristotle]
8. ~p [6, Aristotle]

x

1. Relevance

Connexive logic satisfies Anderson's and Belnap's criterion of relevance:- that in a valid implication, the antecedent should be "relevant" to the consequent. For example, that snow is white is irrelevant to the truth of "If

2+2=4, then 2+2=4”. In general, q→(p→p), though valid in 2-valued logic, is invalid in logics which require variable-sharing between the antecedent and consequent of true implications. In connexive tableaux, relevance is ensured by a system of checking:

~[q→(p→p)] ^
q
~(p→p) ^
p ^
~p ^

The checks attached to each of the lines 1, 3, 4 and 5 of the above tableau indicate that the line in question was <u>used</u> in procuring the contradiction. Line 2 was not used, and “q” stands as irrelevant to the process of closing the tableau. In connexive logic, a tableau at a world does not close unless every line in every path through the tableau is checked.

2. Double checking

Propositional theses such as

[p→(p→q)]→(p→q) (Hilbert) and
[p→(q→r)]→[(p→q)→(p→r)] (Frege),

where one variable occurs an odd number of times and all others occur an even number of times, are also invalid in connexive logic. This is shown by the failure of Hilbert and Frege to satisfy Angell’s 4 x 4 implication/negation matrices for connexive logic (McCall (1966), p. 418). We have the following tableau for Hilbert:

1. ~{[p→(p→q)] → (p→q)} ^
2. p→(p→q) ^ [1]
3. ~(p→q) ^ [1]
4. p ^ ^ [3]
5. ~q ^ [3]
↙ ↘
6. ~p ^   p→q ^ [2]
↙ ↘
7. ~p ^   q ^ [6]

Note that p is checked twice on line 4.

Similarly in the case of Frege:

1. ~{[p→(q→r)] → [(p→q)→(p→r)]} ^
2. p→(q→r) ^ [1]
3. ~[(p→q)→(p→r)] ^ [1]
4. p→q ^ [3]
5. ~(p→r) ^ [3]
6. p ^ ^ [5]
7. ~r ^ [5]

↙ ↘

8. ~p ^   q→r ^ [2]

↙ ↘

9. ~p ^   q ^ [4]

↙ ↘

10. ~q ^   r ^ [8]

Again, p is double-checked on line 6, once to pair with ~p on line 8, and once to pair with ~p on line 9. In connexive tableaux, a path containing a formula that is checked more than once does not close, thus ruling out the non-connexive theses Hilbert and Frege.

3. Fallacies of necessity

In Anderson-Belnap logic, a contingent matter-of-fact proposition p cannot imply an implication q→r, since implications if true at all are true necessarily, not as a matter of contingent fact. Thus although (p→q)→(p→q) holds, the law of assertion p→[(p→q)→q] does not. But if p itself is an implication and hence necessary, all is well, and we have as a theorem P→[(P→q)→q] ("Weak Assertion"), where the capitalized variable stands for an implication. In the latter proposition there is no "fallacy of necessity": a contingent proposition does not imply a necessary one. p→[(p→q)→q] comes from (p→q)→(p→q) by commutation of the antecedents p→q and p, and such commutation is ruled out if in place of [p→(q→r)]→[q→(p→r)] ("Comm") we have as a theorem [p→(Q→r)]→[Q→(p→r)], where capitalized variables P and Q represent only implications, not contingent propositions. [p→(Q→r)]→[Q→(p→r)] may be labelled "Weak Comm". We have, in tableau form, the following for Comm:

```
        1. ~{[p→(q→r)] → [q→(p→r)]} ^
                 2. p→(q→r) ^
                3. ~[q→(p→r)] ^
               4. q   [from line 3]
                 5. ~(p→r) ^
              6. p ^   [from line 5]
                    7. ~r
                    ↙   ↘
            8. ~p ^         q→r ^   [from line 2]
             x              ↙   ↘
                     9. ~q        r
                                  x
```

This tableau is prevented from closing by the fact that between q and ~q there lies (on the same branch) ~(p→r). We have as a rule that contradictory pairs consisting of variables and their negations which lie within the same branch cannot be checked if they are separated (also in the same branch) by formulae containing a negated arrow. This is the case with q on line 4, which is separated from ~q on line 9 by ~(p→r). Consequently the tableau does not close. In connexive tableaux, the <u>ordering</u> of the items in a tableau is crucial.

Consider now Weak Comm, i.e. [p→(Q→r)] → [Q→(p→r)], written in the form [p→[(q→r)→s)]] → [(q→r)→(p→s)]:

```
      1. ~{[p→[(q→r)→s)]] → [(q→r)→(p→s)]}
             2. p→[(q→r)→s] ^   [1]
            3. ~[(q→r)→(p→s)] ^   [1]
                 4. q→r ^   [3]
                5. ~(p→s) ^   [3]
                   6. p ^   [5]
                   7. ~s ^   [5]
                  ↙        ↘
       8. ~p ^          (q→r)→s ^   [2]
        x                ↙        ↘
              9. ~(q→r) ^         s ^   [8]
             10. q ^                 x   [9]
             11. ~r ^   [9]
              ↙        ↘
        12. ~q ^        r ^   [4]
             x          x
```

In this tableau, starting from the left, the first branch closes with p and ~p, the second with q and ~q, the third with ~r and r, and the fourth with ~s and s. These pairs are not separated by formulae containing a negated arrow. When Comm is replaced by Weak Comm, fallacies of necessity are avoided.

An alternative and more perspicuous way of bringing about the closure of the Weak Comm tableau, while leaving the Comm tableau open, is to adopt a tableau “rule of reiteration”, which permits the transfer of an implication from a higher to a lower position on a branch, while forbidding the transfer of a negated implication or a propositional variable. Suppose we adopt a new Anderson-Belnap type rule similar to their “rule of reiteration” in subproof formulations, namely that any implication of the form A→B may be reiterated from a subproof X to an inner subproof Y which is subordinate to X. For semantic tableaux, the rule is:

Tableau Rule of Reiteration. Any implication of the form A→B, where A and B are wffs, may be repeated from its position in a tableau T to a position lower down on some branch or branches of T.

Applying this rule to the tableau for Weak Comm, we get:

```
          1. ~{[p→(Q→r)] → [Q→(p→r)]} ^
                 2. p→(Q→r) ^   [1]
               3. ~[Q→(p→r)] ^   [1]
                     4. Q ^   [3]
                 5. ~(p→r) ^   [3]
                     6. p ^   [5]
                    7. ~r ^   [5]
                    ↙        ↘
          8. ~p ^          Q→r ^   [2]
              x            ↙        ↘
                    9. ~Q ^         r ^   [8]
                    10. Q ^           x   [4, Tableau Reiteration]
                         x
```

4. Conjunction and Disjunction

There are straightforward tableau rules for conjunction and disjunction:

A&B	~(A&B)	A∨B	~(A∨B)
A	↙ ↘	↙ ↘	~A
B	~A ~B	A B	~B

Here are some typical tableaux:

1. ~{(p→q) → [(p&r)→(q&r)]} ^
2. p→q ^ [1]
3. ~[(p&r)→(q&r)] ^ [1]
4. p&r ^ [3]
5. ~(q&r) ^ [3]
6. p ^ [4]
7. r ^ [4]

↙ ↘

8. ~q ^ ~r ^ [5]

↙ ↘ x

9. ~p ^ q ^ [2]

x x

1. ~[(p∨q) → ~(~p&~q)] ^
2. p∨q ^ [1]
3. ~~(~p&~q) ^ [1]
4. ~p&~q ^ [3 DN]
5. ~p ^ [4]
6. ~q ^ [4]

↙ ↘

7. p ^ q ^ [2]

x x

Note that we used the rule for double negation at line 4.

1. ~[~(~p&~q) → (p∨q)] ^
2. ~(~p&~q) ^ [1]
3. ~(p∨q) ^ [1]
4. ~p ^ [3]
5. ~q ^ [3]

↙ ↘

6. ~~p ^ ~~q ^ [2]

x x

In the above tableau, we operated on line 3 before operating on line 2. If however we had operated on 2 before 3, the tableau would not have closed:

1. ~[~(~p&~q) → (p∨q)] ^
2. ~(~p&~q) ^ [1]
3. ~(p∨q) ^ [1]
↙ ↘
4. ~~p ^ ~~q ^ [2]
5. p ^ q ^ [4]
6. ~p ^ ~p [3]
7. ~q ~q ^ [3]

This tableau does not close because ~q is not checked on the left, and ~p is not checked on the right.

Exportation

1. ~{[(p&q)→r] → [p→(q→r)]}
2. (p&q)→r ^ [1]
3. ~[p→(q→r)] ^ [1]
4. p [3]
5. ~(q→r) ^ [3]
6. q ^ [5]
7. ~r ^ [5]
↙ ↘
8. ~(p&q) ^ r ^ [2]
↙ ↘ x
9. ~p ~q ^ [8]
x

No closure because ~(q→r) lies between p and ~p. Hence p and ~p cannot be checked. If however p is replaced by the implication variable P, yielding Weak Exportation [(P&q)→r] → [P→(q→r)], then P can be reiterated from line 4 to the line below line 9, and the tableau closes.

1. ~~(p&~p) ^
2. (p&~p) ^ [1 DN]
3. p ^ [2]
4. ~p ^ [2]
x

However, (p&q)→p, (p&p)→p, and p→(p&p) all lead to open tableaux:

1. ~[(p&q)→p] ^
2. p&q ^ [1]
3. ~p ^ [1]
4. p ^ [2]
5. q [2]

q is unchecked.

1. ~[(p&p)→p] ^
2. p&p ^ [1]
3. ~p ^ ^ [1]
4. p ^ [2]
5. p ^ [2]

~p is double checked.

1. ~[p→(p&p)] ^
2. p ^ ^ [1]
3. ~(p&p) ^ [1]
↙ ↘
4. ~p ^ ~p ^ [3]

p is double checked.

In the case of (p&p)→p and p→(p&p), these formulae fail to satisfy the connexive 4x4 truth-matrices for connexive implication found in McCall (1966), p. 418, where the conjunction matrix should read as follows (values 1 and 2 being "designated"):

&	1	2	3	4
1	1	2	3	4
2	2	1	4	3
3	3	4	3	4
4	4	3	4	3

However, there are other versions of connexive logic, including a Gentzen sequent formulation, in which (p&p)→p and p→(p&p) *are* provable, and in the future it may be possible to devise tableau rules which satisfy these formulae.

5. Completeness of the tableau method

To prove completeness of the tableau method for connexive implication, we must first of all give closed semantic tableaux for each of the connexive axioms given in McCall (1966), pp. 425-426. Start with the pure implicational fragment of connexive logic, which has as axioms Syl and [(p→p)→q]→q. The tableau for Syl was given above, and for [(p→p)→q]→q we have:

```
1. ~{[(p→p)→q]→q} ^
   2. (p→p)→q ^   [1]
      3. ~q ^   [1]
        ↙        ↘
4. ~(p→p) ^          q ^   [2]
 5. p ^                x   [4]
6. ~p ^   [4]
    x
```

From Syl and [(p→p)→q]→q we derive p→p (see McCall (1966), p. 426). Moving from the pure implicational fragment of connexive logic to the implication-negation part, we add "Aristotle" and "Boethius" (see above, pp. 1 and 2). Conjunction and disjunction axioms are found in McCall (1966).

To prove completeness of the tableau method, we first construct closed semantic tableaux for each of the connexive axioms. Then we must show that any theorem that can be obtained from the axioms by use of the rules of substitution, detachment (modus ponens), and adjunction (from |- A and |- B derive |- A&B), also generates a closed tableau. Substitution is no problem. That the rule of detachment leads from closed tableaux to closed tableaux may be shown as follows.

Suppose we have a closed tableau for A, and a closed tableau for A→B. This means that the tableau headed by ~A closes, and the tableau headed by ~(A→B) closes:

~A   ~(A→B)
.   .
.   .
x   x

We need to show that there is a tableau headed by ~B which closes, i.e.

~B
.
x

It is not difficult to construct such a tableau, based on reiteration of the classical two-valued theorem [(A→B) & A] → B:

1. ~B ^
2. [(A→B) & A] → B ^ [Reiteration of theorem]
↙ ↘

3. ~[(A→B) & A] ^   B ^ [2]
↙ ↘   x

4. ~(A→B)   ~A [3]
.   .
.   .
x   x

In this tableau, the left branch closes because it was originally assumed that there existed a closed tableau for A→B, and the middle branch closes because of the assumption that there was a closed tableau for A. The right branch closes because both B and ~B occur on the path. Consequently the entire tableau closes, i.e. there exists a closed tableau for B. Use of the rule of detachment in proving theorems from connexive axioms leads from closed tableaux to closed tableaux. If there is a closed tableau for formula X, and a closed tableau for X→Y, then there is a closed tableau for Y.

Finally, the rule of adjunction is justified by detachment plus reiteration of the theorem A → (B → (A&B)). The set of all closed semantic tableaux is complete. ■

6. Soundness of the tableau method

The 4-valued truth-matrices for connexive implication found in McCall (1966) include the amended conjunction matrix of section 4, and the following implication-negation one:

→	1	2	3	4	~
1	1	4	3	4	4
2	4	1	4	3	3
3	1	4	1	4	2
4	4	1	4	1	1

&	1	2	3	4
1	1	2	3	4
2	2	1	4	3
3	3	4	3	4
4	4	3	4	3

As before, the values 1 and 2 are designated. We shall show that the system of semantic tableaux rules for connexive logic is sound in the sense that all formulae provable using the rules satisfy the matrices, i.e. are four-valued tautologies in the sense of receiving either a 1 or a 2 for all values of their variables. If this is so, use of the tableau rules never leads to a contradiction.

Let v be a propositional interpretation which, starting with assignments to variables, takes every wff into one of the four values {1, 2, 3, 4}. Let b be any branch of a tableau that is being constructed, and let v be faithful to the branch b iff for every wff A on branch b, v(A) = 1 or 2.

Soundness lemma. (I am indebted to Priest (2001), p.15 ff., for the structure of this soundness proof.) If v is faithful to a branch b, and if a tableau rule (not including the rule Aristotle) is applied to b, then v is faithful to at least one of the branches generated by the application of the tableau rule.

Proof. Proof consists of a case-by-case examination of the tableau rules. I shall examine only the rules for implication; the rules for conjunction and disjunction are treated in a similar fashion.

Let A→B occur on branch b. Then b splits into two branches b' and b", with ~A on b' and B on b":

A→B
↙ ↘
~A B

Since A→B is on b, and since v is faithful to b, then v(A→B) = 1 or 2. But v(A→B) = 2 is impossible, since the matrix for → contains no instances of the value 2. Hence v(A→B) = 1, and either (i) v(A) = v(B), or (ii) v(A) = 3 and v(B) = 1, or (iii) v(A) = 4 and v(B) = 2.

v(A) = v(B). There are four subcases:

(ia) v(A) = 1. Then v(B) = 1, and v is faithful to branch b"
(ib) v(A) = 2. Then v(B) = 2, and v is faithful to branch b"
(ic) v(A) = 3. Then v(~A) = 2, and v is faithful to branch b'
(id) v(A) = 4. Then v(~A) = 1, and v is faithful to branch b'.

v(A) = 3 and v(B) = 1. Then v is faithful to branch b".

v(A) = 4 and v(B) = 2. Then b is faithful to branch b".

Hence in all possible cases, v is faithful to one or other of the two branches b' and b".

Let ~(A→B) occur on branch b. Then b does not split:

~(A→B)
A
~B

Since v is faithful to b, then v(~(A→B)) = 1 or 2.

Subcase 1 v(~(A→B)) =1.

Subcase 1.1 v(A) = 1 and v(B) = 2. Hence v(~B) = 3.
This case is impossible, since v is not faithful to branch b.
Subcase 1.2 v(A) = 1 and v(B) = 4. Hence v(~B) =1.
In this case, v is faithful to branch b.
Subcase 1.3 v(A) = 2 and v(B) = 1. In this case v(~B) = 4.

This case is impossible, since v is not faithful to branch b.

Subcase 1.4 v(A) = 2 and v(B) = 3. Consequently v(~B) = 2, and v is faithful to branch b.

Subcase 1.5 v(A) = 3 and v(B) = 2.

1.6 v(A) = 3 and v(B) = 4.

1.7 v(A) = 4 and v(B) = 1.

1.8 v(A) = 4 and v(B) = 3.

In none of these subcases 1.5 to 1.8 is v faithful to branch b, because v(A) = 3 or 4. Consequently all of these subcases are impossible.

Subcase 2 v(~(A→B)) = 2.

It follows that v(A→B) = 3, and there are two subcases.

Subcase 2.1 v(A) = 1 and v(B) = 3. Then v(~B) = 2, and v is faithful to branch b.

Subcase 2.2 v(A) = 2 and v(B) = 4. Then v(~B) = 1, and v is faithful to branch b.

This concludes the proof of the soundness lemma. ■

Soundness theorem

Let |- A mean that A is derivable using the tableau method, and let |= A mean that A is a four-valued tautology. Then the soundness theorem states that:

If |- A, then |= A.

Proof. Assume |- A. Then, by the completeness of the tableau method there is a closed tableau T for ~A. Let v be a propositional interpretation that takes every line of T into one of the four values {1, 2, 3, 4}, and let v(~A) = 1 or 2. Let the initial branch of the tableau, on which ~A is located, be named "b". Since v(~A) = 1 or 2, v is faithful to b at level 1, the top of the tableau.

By the soundness lemma, if v is faithful to b at level 1, v must be faithful to b at each subsequent level, i.e. at levels 2, 3, 4, … . But this is impossible. Tableau T is closed, and every branch of T contains either two mutually contradictory items B and ~B introduced by the rule Aristotle, or at least one propositional variable and its negation. (If the former, the latter follows.) If v(p) = 1 or 2, v(~p) = 3 or 4, and v cannot be faithful to any of the branches of T. The assumption that v(~A) = 1 or 2 leads to a contradiction, and consequently v(~A) = 3 or 4. But then v(A) always equals 1 or 2, and it follows that |= A. ■

7. Conclusion

Semantic tableau rules for connexive logic have been introduced, and the set of formulae derivable using these rules has been shown to be sound and complete.

References

Angell, R. B. (1962). A propositional logic with subjunctive conditionals. *The Journal of Symbolic Logic* (JSL) 27, pp. 327-343.

Jeffrey, R. C. (1967). *Formal Logic: Its Scope and Limits*. McGraw-Hill, New York.

McCall, S. (1966). Connexive implication. *JSL* 31, pp. 415-433.

------------ (2012). A history of connexivity. In Gabbay, Pelletier and Woods (eds.) *Handbook of the History of Logic*, vol. 11, pp. 415-449.

------------ (2014). Connexive Gentzen. *Logic Journal of the IGPL* 22, pp. 964-981.

Priest, G. (2001). *An Introduction to Non-Classical Logic*. Cambridge: Cambridge University Press.

Wansing, H. (2014). Connexive logic. In N. Zalta (ed.). *Stanford Encyclopedia of Philosophy*. Retrieved from <http://plato.stanford.edu/entries/logic-connexive/>.

Storrs McCall
Department of Philosophy
McGill University
storrs.mccall@mcgill.ca