The Game of Tableau

For the time being, I’d like you to set aside everything you know or have heard so far about the meaning or semantics of our logical language. We’re going to play a little game I like to call Tableau. The object and the rules of this little game are quite simple.

1. Start by drawing a vertical line. This will be the root path. On the left of this line, you may place any set of formulas, \( \Gamma \) (including an empty set). You may, if you like, place another formula, \( \varphi \), or set of formulas, on the right of this line.

   \[
   \Gamma \mid \varphi
   \]

2. We develop a tableau by choosing compound formulas along an open path and then extending those paths according to the rules corresponding to those formulas. These rules are detailed below (see 4.), and will sometimes require one to split (or branch) an open path into two. Separate paths trace their ways back up to the root. A path on a tableau closes (or a branch dies!) just in case some formula (any formula) appears both on the left and on the right of that path. One does not develop a closed path any further. We indicate that a path has closed by drawing an ‘X’ below it:

   \[
   \varphi \mid \varphi
   \]
   \[
   \varphi
   \]
   \[
   \varphi
   \]
   \[
   X
   \]

   Note that these formulas may be separated from one another by other formulas, and also by multiple branchings.

3. The object of this game is to create, if possible, a closed tableau – one in which every path has been closed. We can symbolize this situation as follows:

   \[
   \Gamma \mid \varnothing
   \]

   Once again, this basically means that a closed tableau can be constructed with \( \Gamma \) on the left of the root and \( \varnothing \) on the right.

4. The rules for developing a tableau apply to any open path below a formula, and some require paths to split (or to branch). What the rules direct you to do depends upon the
type of formula to which the rule is being applied and whether that formula appears on
the right or the left side of a given path.

\[
\begin{array}{|c|c|c|c|}
\hline
\neg \phi & \neg \phi & \phi \land \psi & \phi \land \psi \\
\phi & \phi & \phi \lor \psi & \phi \lor \psi \\
\hline
\end{array}
\]

5. A tableau path is fully developed just in case it is either closed or one cannot apply any
more rules of development on it without duplication. An entire tableau is fully developed
just in case all of its paths have been fully developed. Since all of these rules decompose
formulas into their constituents we can be sure that all such tableau (at least all that start
out with a finite set of formulas in their initial setup) can be fully developed.

6. Since the object of the game is to close off paths as quickly as possible (or – more
colorfully – to “kill branches”) it is usually advisable to develop formulas with an eye to
“killing” off branches as soon as possible, and also to develop formulas that don’t branch
before developing those that do. It’s easier to prune palm trees than to hack away at
kudzu vines.

That’s all there is to the game of Tableau. Have fun playing!

**Exercises**

Establish the correctness of the following claims involving |--.

1. (The rule of distribution)
   a. A&(BvC) |-- (A&B) v (A&C)
   b. (A&B) v (A&C) |-- A&(BvC)

   [Note that this is really two claims that can be represented as A&(BvC) |--|-- (A&B) v
    (A&C)
2. \( P \rightarrow \neg (P \rightarrow Q) \implies P \)

3. \( P \lor Q \rightarrow \neg (P \rightarrow Q) \implies (P \rightarrow Q) \rightarrow Q \)

4. \( \neg (P \lor \neg P) \) [The Law of Excluded Middle]

5. \( \neg (P \rightarrow (P \lor Q)) \)

6. \( \neg (P \lor Q) \land (\neg P \lor \neg Q) \)

7. \( \neg (((P \rightarrow Q) \rightarrow P) \rightarrow P) \) [This is sometimes called “Peirce’s Law”]

8. \( (P \land \neg (Q \lor P)) \)

9. \( \neg (Q \lor P) \)

10. \( \neg (P \lor Q) \land (\neg P \lor \neg Q) \)

11. \( (P \rightarrow Q), (Q \rightarrow R), (R \rightarrow S) \implies (P \rightarrow S) \)

12. \( (P \lor Q), \neg (P \land R), \neg (Q \land S) \implies \neg (R \land S) \)

13. \( (P \rightarrow (Q \lor R)), (R \rightarrow (P \rightarrow S)), \neg (S \land P) \implies (P \rightarrow Q) \)

14. \( (P \rightarrow (Q \lor R)), (R \rightarrow (P \rightarrow S)), \neg (S \land P) \implies (Q \rightarrow P) \)

[Note that exercises 11-14 above involve the same formulas as exercises 5(a)-(d) from our previous discussion of the semantics of SL.]

15. \( \neg (P \lor Q) \rightarrow \neg (P \lor R) ; \neg (P \lor Q) \rightarrow \neg (Q \lor R) ; (P \lor \neg Q) \rightarrow \neg (R \lor \neg P) \implies P \rightarrow \neg (R \lor \neg Q) \)

16. \( (P \rightarrow Q) \lor (Q \rightarrow R) ; \neg R \rightarrow \neg (P \land Q) \implies (Q \rightarrow \neg P) \)

17. \( (P \land Q) \rightarrow \neg R ; (Q \land R) \rightarrow \neg P ; (P \lor Q) \rightarrow \neg (Q \lor R) \implies (P \land \neg (Q \land R)) \)

18. \( (P \lor Q) \rightarrow \neg (R \land S) ; (R \land P) \rightarrow \neg S ; (P \lor Q) \lor S \implies (S \lor P) \rightarrow \neg (Q \land R) \)

19. \( (P \lor Q) \rightarrow R ; (R \lor S) \rightarrow \neg T \implies (P \lor \neg T) \)

20. \( P \rightarrow Q ; \neg P \lor Q \implies (Q \rightarrow P) \)

21. \( \neg P \lor Q ; \neg (P \lor (S \rightarrow R)) ; Q \lor (R \lor P) \)