Completeness of the Tree System for Propositional Classical Logic

Shahid Rahman

Completeness of the Tree System for Propositional Classical Logic

Shahid Rahman

shahid.rahman@univ-lille3.fr
Univ. Lille; Philosophie, UMR 8163: STL

We already mentioned that when a given proof system is “sound” we mean that with this system we cannot prove any formula it should not. Now, a system could be sound because it proves too “little” or, worse, because it proves anything. We would also like to know that the proof-system proves all what it should. For example; if our tree system were not complete then we would not be able to prove some formula, such as $B \rightarrow (A \rightarrow B)$, which is an element of the set of valid formulae of classical logic. More generally, to say that a proof system is complete in relation to classical logic means: if a formula φ is valid in classical logic**,** then there will a proof (a closed tree) in the corresponding proof system (in our case the Tree system) .

Notice that if the proof system is inconsistent (contradictory), it will be trivially complete: indeed, every formula could be proved. That is why when tackling the issue of completeness it is advisable to check soundness before.

Thus, what we need to prove is that if a formula is classical valid there will be a closed tree for this formula. In fact, we will follow the standard practice to prove the contrapositive: for any formula, if there is no closed tree for it, then there is a model in which that formula fails to be true, so that formula is not valid. The ides is to prove this by showing how to extract a countermodel from a failed attempt to develop a proof.

1 Preparatives

1.1 Worked out trees

Let us say that a tree has been *worked out* if all appropriate tree-rule applications have been made.

1.1.1 Systematically worked out trees:

What we need is a systematic method for constructing a tree that ensures that if we start a tree for a given formula, we can always produce a worked out a tree. There are many options available the following, as we will discuss below,. Now, systematic worked out trees do not contribute to no insight it is only a mechanical procedure:

1. First stage: thesis

2. Any other stage:
   2.1 pick up the leftmost open branch
   2.2 pick up a formula that is not an atomic formula and do the following going from top to bottom:

2.2.1 If it is a formula of the form **F**-~φ, simply apply the appropriate rule to it. In such a way that the resulting subformula will be added to the end of each open branch on which this negation occurs only if it does not occur there before. Similar for **T**-~φ,

2.2.2 If the formula is a **T**-conjunction or a **F**-disjunction add both of the corresponding subformulae to the end of each open branch on which the conjunction/disjunction occurs only if it does not occur there before. Similar for a **F**-conditional

2.2.3 If the formula is a **T**-conjunction **or a F**-disjunction split the end of each open branch on which the conjunction/disjunction occurs and add the corresponding subformulae only if it does not occur there before. Similar for a **T**-conditional

(Tick any formula that has been subject of the application of a rule)

After all this has been done, do the same for the second from the left and so on.

- Notice that systematic dialogical trees avoid the repetitions.
- Notice that it does method leave it open in some cases what to chose actually it does not matter but we might use the idea to giver priority to the application of non-branching rules over the branching ones.

1.2 Definition 1 [Satisfiable]:

Let us consider a systematically ***worked out*** tree with the open branch **B**. We show how to construct a model ***M***,*v* in which **B** is satisfiable.

For every atomic formula *A*:

1. If **T***A* occurs on **B**, where *A* is an atomic formula, take *v*(*A*)=1 in ***M***
2. If **F***A* occurs on **B**, where *A* is an atomic formula, take *v*(*A*)=0 in ***M***
3. If neither **T***A* nor **T***A* occur on **B**, where *A* is an atomic formula, take *v*(*A*)=0 in ***M***

(recall that the branch has been worked out; so that if there are complex formulae in the branch; the corresponding atomic subformulae will also occur)

2 Proofs

2.1 2.2 Completeness lemma

THESIS:
For *each* formula *Φ* (atomic or not) on the open branch **B** of a given systematically ***worked out*** tree we can determine a model ***M,****v* in the following way:

- If **T**Φ occurs on **B**,then $v(\Phi)=1$ in $\boldsymbol{M}$
- If **F**Φ occurs on **B**, $v(\Phi)=0$ in $\boldsymbol{M}$

PROOF:
By induction on the complexity of the formula Φ.

> **Reminder**
>
> - *Induction proof*: This is used in many proofs in mathematics and in logic. For our purposes the proof amounts to the following:
> - Suppose that some property $\mathbb{P}$ can be predicated of all atomic formulae (this is called the *base case*).
> - And suppose that we can prove that whenever one constructs complex formulae out of formulae of which P can be predicated, the resulting formulae also have that property (this is called the *induction clause*).
> - Then it follows that all the formulae of the language have $\mathbb{P}$.
> - The proof of the induction clause breaks down normally into a number of different cases, for our purposes: one for each of the connectives employed in the construction of complex formulae. E;g; if we assume A has $\mathbb{P}$ then $\neg A$ too; if A and B have $\mathbb{P}$ then $A\to B$ too etc. Each of the assumptions in the proof is called *inductive hypothesis*.

The property at stake in our case is the one described in the thesis

BASE CASE: Assume Φ is the atomic formula A

1) If **T**A occurs on **B**, where A is an atomic formula, by point 1 of definition 1 $v(A)=1$ in $\boldsymbol{M}$

   If **F**A occurs on **B**, where A is an atomic formula, **T**A does not occur on the branch, since is open. Thus, by point 2 of definition 1 $v(A)=0$ in $\boldsymbol{M}$**.**

2) INDUCTION CLAUSE

Assume (induction hypothesis) that we know the result for formulae simpler than Φ.,

Let us start with the case where Φ is $A\to B$. If **F**$(A\to B)$ occurs on **B**, since the tree has been worked out, the following formulae occur on **B**, too:

TA
FB

Since, these are simpler than Φ, by induction hypothesis we will have that:

$v(A)=1$ in $\boldsymbol{M}$. and
$v(B)=0$ in $\boldsymbol{M}$.

Thus, as required

$v(A \rightarrow B)=0$ in ***M***

The other conjunctive cases are similar (that is cases where the application of the particle-rule does not yield a branching)

I leave the rest of the cases to the reader

Completeness Proof for propositional Classical Logic:

THESIS:

If *A* is classically valid , then there is a proof for it using tree rules for classical logic. If there is such a proof this proof amounts to the construction of a closed tree for **F**A.

We prove the contrapositive:

If there no tree-proof for *A* then, *A* is not valid.

If we start the proof with **F***A*, and work out a systematic, it should not close (since by assumption of the contraposition there is no proof for *A).*

Let us pick up the open branch **B** of the systematic tree. With help of the completeness lemma we can create a model for which **F***A* holds.

In particular, since **F***A* occurs on **B**, then there is model ***M*,**, at which Φ is false. In our case, Φ is *A*, and **F***A* is the start of the systematic tree, so it is on every open branch. Thus there is a model; at which *A* is false, so it is not valid.

Quod erat demonstrandum

logic.