teorema
Vol. XLII/1, 2023, pp. 123-147
ISNN 0210-1602
[BIBLID 0210-1602 (2023) 42:1; pp. 123-147

The Truth Table Formulation of Propositional Logic

Tristan Grøtvedt Haze

RESUMEN

Desarrollando una sugerencia de Wittgenstein, ofrezco una explicación de las tablas de verdad como fórmulas de un lenguaje formal. Defino la sintaxis y la semántica de TPL (el lenguaje de la lógica proposicional tabular) y desarrollo su teoría de la demostración. Las fórmulas individuales de TPL y los grupos finitos de fórmulas con la misma fila superior y matriz TF (representación de posibles valoraciones) pueden servir como sus propias pruebas con respecto a las propiedades metalógicas de interés. Sin embargo, la situación es diferente para los grupos de fórmulas cuyas filas superiores difieren.

PALABRAS CLAVE: *lógica proposicional, tablas de verdad, filosofía de la lógica, simbolismo, notación, demostración formal.*

ABSTRACT

Developing a suggestion of Wittgenstein, I provide an account of truth tables as formulas of a formal language. I define the syntax and semantics of TPL (the language of Tabular Propositional Logic) and develop its proof theory. Single formulas of TPL, and finite groups of formulas with the same top row and TF matrix (depiction of possible valuations), are able to serve as their own proofs with respect to metalogical properties of interest. The situation is different, however, for groups of formulas whose top rows differ.

KEYWORDS: *Propositional Logic, Truth Tables, Philosophy of Logic, Symbolism, Notation, Formal Proof.*

I. INTRODUCTION

In logic, truth tables appear in multiple aspects:

Defining connectives. Truth tables are used to define the connectives that appear in PL wffs (i.e., standard formulas of propositional logic).[1]

Defining metalogical notions. In the setting of propositional logic, truth tables are used to precisify logical notions such as those of tautologousness (a formula is a *tautology* iff it is true on every row of its truth table), validity (an argument is *valid* iff there is no truth table row on which its premises are true and its conclusion false), equivalence (two formulas are *equivalent* iff they get the same truth-value on every row), and so on. Here, truth table rows play the role of models.

Deciding presence of metalogical notions. Truth tables constitute an effective decision procedure for the metalogical properties mentioned above – they are finite arrays composed of a definite stock of symbols and can be generated mechanically in a finite amount of time.

Proving presence of metalogical notions. A completed truth table for a PL wff constitutes a proof of that wff's tautologousness (if it is a tautology), satisfiability (if it is satisfiable), or unsatisfiability (if it is unsatisfiable), since truth tables can be checked mechanically for correctness in a feasible time, relative to the size of the table (i.e. the time it takes to check truth tables does not get out of hand as the tables get larger).

But truth tables have yet another side to them, highlighted by Wittgenstein; what we might normally call the truth table *for* some particular formula of propositional logic in some standard notation may also be regarded as a formula itself.[2]

For example, consider the following truth table for the PL wff $(p \rightarrow q)$:

p	q	$(p \rightarrow q)$
T	T	T
T	F	F
F	T	T
F	F	T

If we got rid of the PL wff in the top right, the remainder could be used in its stead.

In this paper I work out a systematic account of this aspect of truth tables using modern logical methods. Why do this? Well, I hope that the interest of some students of logic has already been kindled by the very idea of doing this, but I can say more. I have three main motivations for working out such an account:

Motivation 1: Broadening the Church of Modern Logic. I would like to see more attempts to apply modern logical techniques to structures that resemble natural language sentences less than do formulas of standard logical languages. And I am not talking about liberating the tools of logic from the subject's traditional focus on truth and on what follows from what (although that can be interesting and worthwhile too). Thinking of logic as primarily being about representations and how truth distributes over them – a natural and common view, which I will adopt here but will not argue for – patently does *not* entail a focus on sentence-like representations. By regarding truth-tables as "well-formed formulas" and working out their logic, I hope to loosen things up and help to encourage further logical study of items that are representational without being sentence-like.

Motivation 2: Vindicating Wittgenstein's Idea in an Ecumenical Setting. Wittgenstein had many very interesting ideas about logic, but I think it's safe to say that many of them have, so far at least, failed to catch on. And this idea that truth-tables may be seen as formulas – as propositional signs – may seem quite strange and exotic. Lest it be thought that this exotic-seeming idea only "works" or makes sense in the context of Wittgenstein's own early thinking in all its glory and difficulty, I want to show that it can be cherry-picked and worked out thoroughly in a modern setting. So, the project here is very different from one of working out what the early Wittgenstein thought, or how he should have elaborated his framework of thinking as a whole. The early Wittgenstein may for instance have thought that the soundness and completeness proofs I provide below belong to an inherently illegitimate or confused way of thinking. But in a way that's the point: we don't have to share all of Wittgenstein's scruples in order to work out, for ourselves, his idea about truth-tables. (And of course, none of this is to say that we don't stand to learn anything by thinking about those scruples on another occasion, or that there is no truth in them.)

Motivation 3: Appreciating the Trade-Offs. I want to understand the trade-offs involved, from a modern logical perspective, in doing propositional logic by treating truth tables as formulas. To anticipate, we will see that the problem of assessing a formula for satisfiability becomes trivial, but the problem of assessing multiple formulas for joint satisfiability does not. Relatedly, when working with standard notation for propositional logic it is trivial to form a *semantic conjunction* of two or more formulas – a formula which is true iff the input formulas are true – but this is not trivial in truth table notation.

The plan for the rest of the paper is as follows. In Section II, I specify the language of tabular propositional logic (TPL) and remark on its status as a language. In Section III, I treat the semantics of TPL by defining truth on a model (where a model is a propositional valuation), and note some facts. In Section IV I turn to proof theory. We will see that tautologous, satisfiable, and unsatisfiable formulas constitute proofs of their own tautologousness, satisfiability, and unsatisfiability respectively. On the other hand, jointly satisfiable pairs or larger groups of formulas do not seem to constitute proofs of their own joint satisfiability (and likewise for joint unsatisfiability). Similarly, arguments in the language of TPL with one or more premise do not count as proofs of their own validity. At least, not in general. We will see that, for groups of formulas which have the same top row and TF matrix (i.e., the same arrangement of T's and F's to depict the possible valuations of the atoms), this self-proving property is present. For cases where the self-proving property is arguably not present, I offer two responses: (i) a system of 'row tree proofs', the soundness and completeness of which is then shown, and (ii) a re-writing strategy. It may be wondered why I provide two different responses here. The reason is that, in this paper, I am not concerned to put forward a view about how best to do propositional logic, or anything like that. Rather, I am trying to explore the space of possibilities and come to an appreciation of the trade-offs between different ways of going.

II. THE LANGUAGE OF TPL

II.1. *Symbols of TPL*

Atoms: a, b, …, z, a_1, a_1, …, z_1, a_2, b_2, …, z_2, …

Value signs: T, F

The stopper: ●

Remark on the Atoms: unlike in standard PL syntax, atoms of TPL do not themselves count as wffs, and are not treated as true or false on models. They may be thought of as signs representing conditions which may or may not be met, but which do not *say* that these conditions are met. Compare a natural language phrase like 'whether the sky is blue'. If we think of the atoms this way, then the T's and F's which appear below them should not be thought of as ascribing truth and falsity to the atoms.[3] Alternatively, the atoms may be thought of as standing in for sentences of some language on which TPL is parasitic — much as one might think of the atoms in a standard PL wff for which a glossary has been given. In that case, the T's and F's below them may be thought of as ascribing truth and falsity.

Remark on the Stopper: the purpose of the stopper is to occupy the place where, in a truth table *for* a standard PL wff, that wff would go. The stopper is not really necessary, but having it enables us to treat formulas of TPL as regular two-dimensional arrays of symbols whose size and shape can be specified with two natural numbers.

II.2. *Syntax of TPL*

For all natural numbers n and m, an n-by-m array[4] of symbols of TPL – where n is the number of rows and m is the number of columns – is a *formula* of TPL. For example, the following are formulas of TPL:

$$\begin{array}{ccc} \mathrm{T} & a & \bullet \\ \bullet & \bullet & \mathrm{F} \end{array}$$

$$\begin{array}{cccc} \mathrm{T} & \mathrm{T} & g & \mathrm{F} \\ \bullet & z & \bullet & \bullet \end{array}$$

$$\begin{array}{c} \mathrm{T} \\ \mathrm{T} \\ \mathrm{F} \\ \mathrm{T} \end{array}$$

$$\begin{array}{cccccccc} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{array}$$

We will now define the *well-formed formulas* or *wffs* of TPL, which form a proper subset of the formulas. First some terminology.

Each element in an n-by-m formula has a unique address given by a pair of natural numbers between 1 and n and 1 and m respectively. For example, element (2,4) in the following 5-by-5 formula is highlighted with an underline (not itself part of the formula):

c	T	●	●	*s*
●	●	F	*z*	F
T	T	T	●	T
R	*r*	●	●	●
T	T	F	*s*	T

The *i*th *row* of an *n*-by-*m* formula *F* is a sequence of length *m* of symbols of TPL, whose 1st element is element (*i*, 1) of *F*, whose 2nd element is element (*i*, 2) of F, and so on.

The *i*th *column* of an *n*-by-*m* formula *F* is a sequence of length *n* of symbols of TPL, whose 1st element is element (1, *i*) of *F*, whose 2nd element is element (2, *i*) of F, and so on.

The *top row* of a formula is its 1st row.

The *rightmost column* of an *n*-by-*m* formula is its *m*th column.

The *last element* of a sequence of length *n* is its *n*th element.

An *alternating TF sequence of n segments of length m* is a sequence of length ($n \times m$) whose first *m* elements are T, its next *m* elements are F, and so on.

For all finite natural numbers *n*, a 2^n-by-(n + 1) formula of TPL, call it α, is a wff iff:

1. Elements 1 … *n* of the top row of α are atoms and element (n + 1) is the stopper.
2. No atom occurs more than once in the top row of α.
3. For all numbers *i* between 1 and *n* inclusive, the *i*th column of α is a sequence consisting of an atom followed by an alternating TF sequence of 2^i segments of length $2^n/2^i$.
4. For all numbers *i* between 2 and 2^n inclusive, element *i* of the right column of α is either T or F.

   No other formula of TPL is a wff.

Conditions 1 and 2 are there to ensure that our truth tables have suitable top rows. Note that while in textbooks truth tables are often presented with the atoms in alphabetical order along the top row, this is not required of a wff of TPL.

Condition 3 ensures that each possible truth-value assignment to the atoms in a truth table appears in a row of the table and that the ordering of the rows follows a standard format.

Condition 4 ensures that the truth table is able to represent its own truth-conditions, by ensuring that each row ends in a T or an F, which comes immediately after the Ts and Fs that will be used to determine a truth-value assignment to the atoms. Note the first element in the right-most column of a wff of TPL will be the stopper, in light of condition 1.

Here, for example, are two wffs of TPL:

p	*q*	●
T	T	T
T	F	F
F	T	T
F	F	T

z	*a*	*r*	●
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	F
F	T	F	T
F	F	T	T
F	F	F	F

We will say that the *negation* of a TPL wff α is the wff β that results from swapping all the T's in α's right column for F's and vice versa.

For example, the negation of our first example of a wff above is:

p	*q*	●
T	T	F
T	F	T
F	T	F
F	F	F

And our first example of a wff above is the negation of *it*, too. (For this reason, we have limited use for the term 'negand' in the context of TPL, since if α is β's negation then β is also α's, with the result that whenever we want to speak of a formula's negand we may just as well call it that formula's 'negation'.)

Remark on the status of TPL as a language: While there may be deep and interesting reasons why languages used by humans tend to be one-dimensional, this does not mean that two-dimensional languages are not possible. It is true that in logic, and related disciplines such as computer science, mathematics and linguistics, a formal language is standardly regarded as a subset of the *sequences* or *strings* or *words* over some *alphabet* (i.e., set of symbols) — and obviously, the *sequence* part of this conception is not a good fit for TPL. However, the key notions of formal language theory can be generalised so that arrays of symbols count as items in a language. (For discussion see Giammarresi & Restivo (1997), the aim of which, to quote from the abstract, is 'to generalize concepts and techniques of formal language theory to two dimensions'.)

III. Semantics of TPL

A *model* of TPL is just like an ordinary model for propositional logic: a function v mapping the atoms to members of the set $\{0, 1\}$.

As a preliminary to defining truth on a model, we define some terms. First, if the ith element of the top row of a wff α is the atom A, we will say that the *atom associated with* the ith element of any row of the wff α is the atom A. (The atom associated with an occurrence of T or F in a TPL wff is the atom directly above it. There is no atom associated with the T's and F's in the right column of a TPL wff.)

With respect to an n-by-m wff α and model v, we will say that α's *v-corresponding row* is the row R such that, for all numbers between 1 and (m - 1) inclusive, the mth element E of R is T iff v(the atom associated with E) = 1. (Intuitively, α's v-corresponding row is the one which represents the atoms involved in α as having the values they have on v.)

We may now define truth on a model: a wff α is true on a model v iff the rightmost element of α's v-corresponding row is T.

We may now give the following standard definitions of metalogical concepts.

A wff α is a *tautology* iff it is true on all models.

A wff α is *satisfiable* iff it is true on at least one model.

A wff α is *contingent* iff it is true on at least one model and it is false on at least one model.

A wff α is *unsatisfiable* iff it is false on all models (i.e. if it is not true on any models).

A set of wffs Γ is *equivalent* iff, on every model, either all members of Γ are true or all members of Γ are false. (Arguably, this is an abuse of language, since it is normally wffs themselves that are said to be 'equivalent', but see the next parenthesis below for what is at least a partial corrective.)

A set of wffs Γ is *satisfiable* iff there is a model on which every member of Γ is true.

A set of wffs Γ is *unsatisfiable* iff there is no model on which every member of Γ is true.

(We will also speak plurally of wffs and call them jointly satisfiable, jointly unsatisfiable, or equivalent.)

A wff α is a *consequence* of a set of wffs Γ iff every model v on which every member of Γ is true is also a model on which α is true. (In this case, we may also say that Γ *implies* α, or write $\Gamma \vDash \alpha$.)

An argument $\alpha_1, \ldots, \alpha_n \therefore \beta$ is *valid* iff its premises $\{\alpha_1, \ldots, \alpha_n\}$ imply its conclusion β.

Before we turn to the proof theory of TPL, note that no two distinct wffs with the same top row are equivalent.

If we were to require that the atoms in the top row of a wff appear in alphabetical order, and that all wffs contain the same atoms, then all wffs would have the same top row, and so no two distinct wffs at all would be equivalent.[5]

IV. Proof Theory

IV.1. *Single Formulas*

A remarkable feature of TPL wffs is that they can serve as their own proofs with respect to metalogical properties such as tautologousness, satisfiability, contingency and unsatisfiability. Let us have a detailed look at this, before considering the situation with pairs, larger groups, or arguments made up of TPL wffs.

> Call a wff α a *proof that* α *is a tautology* iff each element of its rightmost column is T.
>
> Call a wff α a *proof that* α *is satisfiable* iff at least one element of its rightmost column is T.
>
> Call a wff α a *proof that* α *is contingent* iff at least one element of its rightmost column is T and at least one element of its rightmost column is F.
>
> Call a wff α a *proof that* α *is unsatisfiable* iff each element of its rightmost column is F.

So far these are just labels. To justify them, we do two things. First, we note that to inspect an array of symbols and determine whether it is a TPL wff that matches one of the above definitions is a computationally feasible task. (This is relevant because if a "proof" cannot be checked feasibly then its status as a proof is dubious.) Secondly, we prove that the set of TPL wffs considered as a set of proofs is sound and complete with respect to the property of being a tautology. (Soundness and completeness with respect to satisfiability, contingency and unsatisfiability can be proven along similar lines.)

> **Theorem 1.** *Soundness of TPL wffs-as-proofs with respect to the property of being a tautology*. If a wff α is a proof that α is a tautology, then α is a tautology (i.e., is true on all models v). *Proof.* Suppose for the purpose of conditional proof that α is a proof that α is a tautology. By the definition of *proof that* α *is a tautology*, the rightmost element of every row of α other than its first row is T. So for all models v, whichever row R of α is α's v-corresponding row, the rightmost element of R is T. So by the semantics of wffs of TPL, α is true on v. So α is a tautology.
>
> **Theorem 2.** *Completeness of TPL wffs-as-proofs with respect to the property of being a tautology*. If a wff α is a tautology, then α is a proof that α is a tautology. We prove the contrapositive — that if a wff α is not a proof that α is a tautology, then α is not a tautology. Assume that α is not a proof that α is a tautology, i.e., some element of its rightmost column is not T. By Condition 4 of the syntax, that element

must be F. Consider the row R that this element appears on and consider a model v such that R is α's v-corresponding row. By the semantics of TPL wffs, α is false on v, since the rightmost element of R is F and R is α's v-corresponding row. So α is not a tautology.

Note that these results rely on Condition 2 of the syntax, which requires that no atom appears more than once in a wff's top row. If we allowed atoms to appear more than once in a wff's top row, then we could not guarantee that each row of a wff corresponds to a distinct, non-empty set of models. For example, consider the following non-wff:

p	p	●
T	T	F
T	F	T
F	T	T
F	F	F

If we applied our definition of truth on a model to this non-wff then it would be false on all models (since only the TT and FF rows are the v-corresponding rows of some model v and they both end in F), but if it counted as a wff, this formula would (by the definition of 'proof that α is satisfiable') also count as a proof that it is satisfiable. We would then have unsound "proofs".

IV.2. *Multiple Formulas*

We have seen that single formulas of TPL serve as their own proofs with respect to properties like tautologousness, satisfiability, contingency and unsatisfiability. We now consider properties possessed by sets of wffs (and arguments) as defined in Section III.

Call a set Γ of TPL wffs *top-row-identical* iff all wffs $\alpha \in \Gamma$ have the same top row. (In addition to calling sets of wffs top-row-identical, we will also permit ourselves to talk of wffs plurally and call *them* top-row-identical.)

To explain the definition a bit: some wffs are top-row-identical when their top rows involve the same atoms in the same order. (By 'top row', I mean the very top row, where the stopper appears — not the first row of value signs: *that* is the second-from-top row.)

We start by considering sets of top-row-identical wffs, for which the situation is broadly similar to that for single wffs, before discussing the general case of groups of TPL wffs, which poses special difficulties.

IV.2.1. *Top-Row-Identical Formulas*

Call a finite array consisting of all the members of a set Γ of top-row-identical TPL wffs a *proof that* Γ *is equivalent* iff, for every number *i* between 1 and the number of rows of a wff $\alpha \in \Gamma$, either the *i*th element of the rightmost column of every wff $\alpha \in \Gamma$ is T, or the *i*th element of the rightmost column of every wff $\alpha \in \Gamma$ is F.

Call a finite array consisting of all the members of a set Γ of top-row-identical TPL wffs a *proof that* Γ *is satisfiable* iff there is a number *i* between 1 and the number of rows of a wff $\alpha \in \Gamma$ such that, for all wffs $\alpha \in \Gamma$, the *i*th element of α's rightmost column is T.

Call a finite array consisting of all the members of a set Γ of top-row-identical TPL wffs a *proof that* Γ *is unsatisfiable* iff there is no number *i* between 1 and the number of rows of a wff $\alpha \in \Gamma$ such that, for all wffs $\alpha \in \Gamma$, the *i*th element of α's rightmost column is T.

Call a finite array consisting of all the members of a set $\{\alpha_1, \ldots, \alpha_n, \beta\}$ of top-row-identical TPL wffs a *proof that the argument* $\alpha_1, \ldots, \alpha_n \therefore \beta$ *is valid* iff there is no number *i* between 1 and the number of rows of a wff $\alpha \in \Gamma$ such that the *i*th elements of the rightmost columns of $\alpha_1, \ldots, \alpha_n$ are all T while the *i*th element of the rightmost column of β is F. (We will also call this a proof that $\{\alpha_1, \ldots, \alpha_n\} \vDash \beta$.)

To justify these labels, we first note that to inspect an array of symbols and to determine whether it is an array of TPL wffs matching some particular one of the above definitions is a computationally feasible task. Having first checked that our array of symbols is an array of TPL wffs, it is then easy to check whether it is a proof of satisfiability using the following procedure: proceed down the rightmost column of the first wff, and if you hit a T, hop across to the corresponding element of the rightmost column of the next wff and see if it is also a T, and if so, hop across again. If you can get through all the wffs finding a T in that spot each time, then stop and say Yes, the array *is* a proof of satisfiability. Otherwise, go back to the first wff and continue down its rightmost column, repeating the hopping across part of the process if you hit another T. If you get to the end of the rightmost column of the first wff and haven't said Yes, say No.

Now, we prove that the set of arrays of top-row-identical TPL wffs, considered as a set of proofs, is sound and complete with respect to the unsatisfiability of top-row-identical finite sets of wffs. (For the rest of this section, all references to sets of wffs are to be understood as references to top-row-identical finite sets of wffs.) Soundness and completeness with respect to equivalence, satisfiability, and validity can be proven along similar lines.

Theorem 3. *Soundness of arrays of top-row-identical wffs, considered as proofs, with respect to the unsatisfiability of finite top-row-identical sets of wffs.* If there is a proof that Γ is unsatisfiable, then Γ is unsatisfiable (i.e., there is no model v on which all members of Γ are true). *Proof.* We prove the contrapositive: if Γ is satisfiable (i.e., if there *is* a model v on which all $\alpha \in \Gamma$ are true), then there is no proof that Γ is unsatisfiable. Suppose for the purpose of conditional proof that there is a model v on which all members of Γ are true. By the semantics of TPL wffs, all wffs $\alpha \in \Gamma$ are then such that the rightmost element of α's v-corresponding row is T. By the syntax of TPL and the fact that Γ is top-row-identical, these v-corresponding rows all appear in the same place, i.e., there is a number i such that for all wffs $\alpha \in \Gamma$, the ith row of α is α's v-corresponding row. And so, there *is* a number – this same i – such that, for each wff $\alpha \in \Gamma$, the ith element of α's rightmost column (a.k.a. the rightmost element of α's ith row) is T. Hence, by the definition of 'proof that Γ is unsatisfiable', there is no proof that Γ is unsatisfiable.

Theorem 4. *Completeness of arrays of top-row-identical wffs, considered as proofs, with respect to the unsatisfiability of finite top-row-identical sets of wffs.* If Γ is unsatisfiable, there is a proof that Γ is unsatisfiable. We prove the contrapositive: if there is no proof that Γ is unsatisfiable, then Γ is satisfiable. Suppose for the purpose of conditional proof that there is no proof that Γ is unsatisfiable, i.e., an arbitrary array A of wffs $\alpha \in \Gamma$ fails to be a proof that Γ is unsatisfiable. Then, by the definition of 'proof that Γ is unsatisfiable', there *is* a number i such that, for each wff $\alpha \in \Gamma$, the ith element of α's rightmost column is T. That means there is a number i such that for each wff $\alpha \in \Gamma$, the rightmost element of the ith row of α is T. By this fact together with the syntax of TPL and the fact that Γ is top-row-identical, the ith rows of all $\alpha \in \Gamma$ are identical. So, there is a model v such that for all $\alpha \in \Gamma$, the ith row of α is α's v-corresponding row. By the semantics of TPL wffs, all $\alpha \in \Gamma$ are true on this model v, since the rightmost element of the ith row they have in common is T. And so Γ is satisfiable.

Having shown that the proof-theoretic status of arrays of top-row-identical arrays of TPL wffs is broadly similar to that for single TPL wffs, we now turn to the general case — arrays of wffs that may not be top-row-identical.

IV.2.2. *The General Case*

As soon as we venture past top-row-identical sets of wffs, difficulties arise. Even sets of wffs involving the same atoms but in different orders are considerably harder to check for metalogical properties. (To transform a group of such wffs so that they become top-row-identical wffs, we cannot just shuffle columns around, since we need the matrices of the wffs to follow the same pattern.) Further complications arise when different wffs involve different atoms.

IV.2.2.1. *Formal Proofs Should Be Easily Checkable*

In textbooks and the like, the notion of formal proof is often introduced with two requirements: for something to be a formal proof, it needs to be constructed from a definite stock of symbols, and the correctness of its construction needs to be mechanically checkable in a finite amount of time. Hilbert (1918) is an early source for this conception. But these two requirements are not enough in general.[6]

It is easy to see that mechanical decision procedures exist for checking whether sets of TPL wffs (and arguments) possess metalogical properties of interest. But arrays of TPL wffs should not, I think, count in general as *proofs* with respect to these properties.

Compare: given an ordinary PL wff, there is a mechanical procedure that will tell you whether it is a tautology. But we do not count PL tautologies themselves as proofs that they are tautologies.

Why not? Beame and Pitassi (2001) consider exactly this question:

> What exactly is a propositional proof? Cook and Reckhow were possibly the first to make this and related questions precise. (...) Since there are only finitely many truth assignments to check, why not allow the statement itself as a proof? (...) The key observation is that a proof is easy to check, unlike the statement itself [Beame and Pitassi (2001), p. 43].[7]

In the case of arbitrary finite sets of TPL wffs, it isn't easy to check for metalogical properties like joint unsatisfiability, and hence arrays of such jointly unsatisfiable wffs cannot in general be regarded as proofs of the joint unsatisfiability of the wffs involved. If we want proofs here, we need a system for producing quite distinct things which do count as proofs, or – closer perhaps to the spirit of Wittenstein's thinking about these matters – a way of re-writing such wffs so that they *are* top-row identical. In the interest of exploring both options in order to appreciate their relative strengths and weaknesses, in section 4.2.2.2. below I turn to developing a proof system for arbitrary finite sets of TPL formulas, the soundness and completeness of which is then shown in section 4.2.2.3. Following that, I turn in section 4.2.2.4. to a re-writing strategy for reducing the case of arbitrary finite sets of TPL formulas to the top-row identical case.

Before proceeding, it is interesting to note that this difference in the proof-theoretic situation for arbitrary finite sets of wffs as opposed to single wffs is not something that arises in ordinary propositional logic: there the two sorts of problems are straightforwardly reducible to one another, since we may easily take the conjunction of a finite set of wffs and may consider the singleton set of a single wff. By contrast, the problem of forming a semantic conjunction of two or more TPL wffs – a TPL wff that is true on a model iff the given TPL wffs are – is non-trivial.

IV.2.2.2. *Row Tree Proofs*

I will describe the system of row tree proofs by describing a procedure for producing proofs in the system. The strategy is similar to that of tree proofs in ordinary propositional logic: we consider putative ways that the wffs we are interested in could all be true together and see whether any of these putative ways are really possible. If so, we have a proof of joint satisfiability (and can read off a model from it). If not, then we have a proof of joint unsatisfiability.

I will describe the procedure by way of example. Let us show that the following wffs are jointly unsatisfiable:

p	q	●
T	T	T
T	F	F
F	T	T
F	F	T

q	r	●
T	T	T
T	F	F
F	T	T
F	F	T

p	r	●
T	T	F
T	F	T
F	T	F
F	F	F

(Since the last wff is the negation of

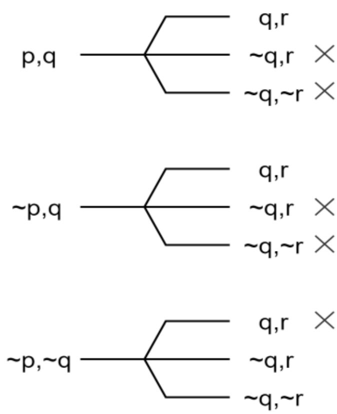
p	r	●
T	T	T
T	F	F
F	T	T
F	F	T

our proof may also be thought of as a proof that a TPL analogue of hypothetical syllogism is valid.)

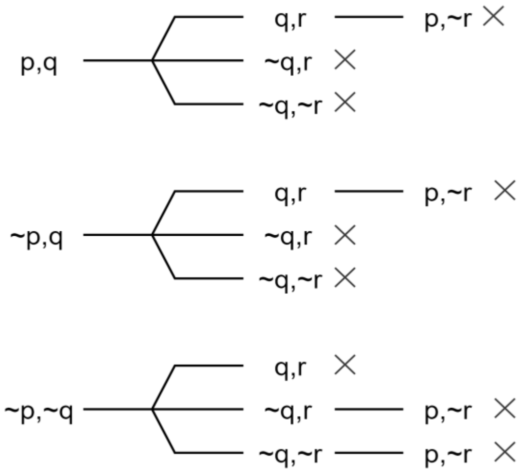
First, we check if any of our wffs have no T's in their rightmost column, and if there are such wffs, we do not begin our construction - these wffs themselves may be regarded as a proof that they are jointly unsatisfiable. Otherwise, we produce a *row tree*, proceeding as follows. We start with our initial wff. For each row R ending with T, we write down R's *row description*: we move along our row from left to right, and when we see a T, we write the atom above the T, and when we see an F, we write the atom above the F but prefixed with a tilde ('~') (call the result the *tilde of* the atom). To ease readability, we may separate these atoms and tilded atoms with commas, but this is not essential.

Now we move on to our next wff. For every row R in this wff that ends in T, we draw a branch coming off each row description from the previous wff and write R's row description. Then, we cross off any path that contains both an atom by itself *and* that atom's tilde.

With our example above, after having considered the first two wffs we should have:



Then, we move on to the next wff and repeat the process of the last step, but only looking at lines in the previous block which have not been crossed off. That gives us:



We continue doing this until all paths close, or until we have dealt with all our wffs. In this example, both of these things just occurred simultaneously — we dealt with our last wff, and in doing so all paths closed.

If all paths close in a completed row tree for a set of TPL wffs, we call the object consisting of the TPL wffs side by side plus the row tree below them a *row tree proof that* Γ *is unsatisfiable*. Otherwise (i.e., if there is at least one open path), we call it a *row tree proof that* Γ *is satisfiable*. So, for our example, we got a row tree proof that the set comprising the wffs we began with is unsatisfiable.

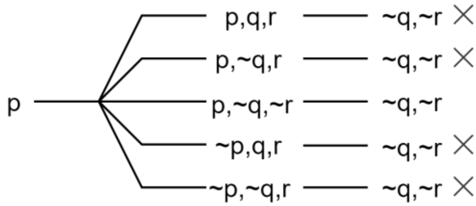
Here is a second example. Consider the following three wffs

p	●
T	T
F	F

p	q	r	●
T	T	T	T
T	T	F	F
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	F
F	F	T	T
F	F	F	F

q	r	●
T	T	F
T	F	F
F	T	F
F	F	T

Here is a row tree which, together with the wffs, constitutes a proof of their joint satisfiability:



From this, we can also read off a model *v* on which our three wffs come out true:

$$v\,(\mathrm{p}) = 1$$
$$v\,(\mathrm{q}) = 0$$
$$v\,(\mathrm{r}) = 0$$

We have now seen how row tree proofs work. It is interesting to note that they correspond to tableaux for ordinary PL wffs written in disjunctive normal form.

Before proving soundness and completeness in the next section, some remarks.

As mentioned in passing earlier, in the case where we do not build a row tree because one of our wffs had no T's in its right column, we may call the array of wffs itself a proof that our set of wffs is unsatisfiable.

We could optimize our procedure a little further by checking, before constructing a row tree, whether any two of our wffs have an atom in common. If not – if our wffs contain disjoint sets of atoms – then we can regard our array of wffs itself as a proof that our set of wffs is jointly satisfiable or unsatisfiable as the case may be: we just check whether each wff has at least one T in its right column. If so, we have a proof of joint satisfiability. If not, a proof of joint unsatisfiability.

A note on the tilde: in standard PL tree proof systems, we often write the negations of formulas of interest. The tilde here plays a similar role as the negation sign plays there, except that it is not part of the language of TPL. Rather, it is an auxiliary sign used only in proofs, like the lines and crosses. Note also that we could have a *signed* variant of the TPL row tree proof system, analogous to signed PL tree proofs, where we don't use the tilde but write a value-sign beside all our atoms. This would be a natural way to extend the TPL row tree proof system for a many-valued logical language. (Note that, unlike with standard many-valued propositional logic, where we use the same language but change its semantics, the natural move to many values for TPL would involve adding more value-signs to the language. Note also that, unlike with standard PL, it is not clear that there is a natural way to extend TPL to infinitely many values.)

IV.2.2.3. *Soundness and Completeness of the Row Tree Proof System*

> **Theorem 5.** *Soundness of row tree proofs with respect to unsatisfiability*. If there is a row tree proof that Γ is unsatisfiable, then Γ is unsatisfiable (i.e., there is no model v on which all members of Γ are true). *Proof.* We prove the contrapositive: if Γ is satisfiable, then there is no row tree proof that Γ is unsatisfiable. For the purpose of conditional proof, assume that there is some model v on which all $\alpha \in \Gamma$ are true. Now, any row tree proof that Γ will be the result of considering the $\alpha \in \Gamma$ in some particular order (at least, until all paths close). For each wff α that we considered, we will have written down (perhaps among other row descriptions) a description of α's v-corresponding row, since α is true on v and so, by the semantics of TPL wffs, α's v-corresponding row ends in a T. But then we will always have at least one open path in our row tree, since no two descriptions of v-corresponding rows for a given v will ever be such that one contains an atom by itself and the other contains that atom's tilde. (This follows by the procedure for describing a row together with the fact that, by the definition of *v-corresponding row*, no two v-corresponding rows for a given v will ever be such that one has a T under some atom and the other has an F under that atom.) Therefore, there is no row tree proof that Γ is unsatisfiable, since by definition all paths close in such a proof.

Before proving completeness, a definition: call a set of wffs Γ *row tree apt* iff it is finite and all $\alpha \in \Gamma$ have at least one T in their right column. Recall that for finite sets of wffs Γ which are not row tree apt, we already have proofs (in the form of the wffs themselves).

> **Theorem 6.** *Completeness of row tree proofs with respect to unsatisfiability*. If a row tree apt set of wffs Γ is unsatisfiable, then there is a row tree proof that Γ is unsatisfiable. *Proof.* We prove the contrapositive: if there is no row tree proof that some row proof apt set of wffs Γ is unsatisfiable, then Γ is satisfiable. For the purpose of conditional

proof, assume the antecedent. That means that all row trees obtainable by considering members of Γ one by one have at least one open path. Consider in particular the finished ones, where all $\alpha \in \Gamma$ have been dealt with. Now consider a model *v* read off one of the open paths *p* in one of these finished row trees. For every $\alpha \in \Gamma$, there is a row description along our path *p*, and the row R of α that yields that description will be α's *v*-corresponding row (by the procedure for reading a model off an open path together with the definition of '*v*-corresponding row'). Row R must end in a T, since otherwise (by the procedure for building a row tree) we would not have written its row description. So, by the semantics of TPL wffs, α is true on *v*. Therefore, all $\alpha \in \Gamma$ are true on *v*, and so Γ is satisfiable.

IV.2.2.4. *The Re-Writing Strategy: Expanding Truth Tables by Means of Redundant Atoms*[8]

First, let us modify clause 1 of the syntax given in 2.2. above to stipulate that the atoms in the top row must come in alphabetical order. Now, consider the following equivalent pair of TPL wffs:

p	*q*	●
T	T	T
T	F	T
F	T	F
F	F	F

p	●
T	T
F	F

In the one on the left, the atom *q* is *redundant*, in the sense that rows which feature the same value sign under each atom except for *q* also feature the same value sign under the stopper. The wff on the left is an *expansion* of the one on the right, in the sense all atoms that appear in the one on the right also appear in the one on the left, and all atoms that appear in the one on the left but not in the one on the right are redundant.

Now, let us return to our example, from 4.2.2.2., of three jointly unsatisfiable wffs:

p	*q*	●
T	T	T
T	F	F
F	T	T
F	F	T

q	*r*	●
T	T	T
T	F	F
F	T	T
F	F	T

p	*r*	●
T	T	F
T	F	T
F	T	F
F	F	F

It is not easy to check whether they are jointly unsatisfiably simply by looking at them. However, we can expand them so that they become top-row-identical:

p	q	r	●	p	q	r	●	p	q	r	●
T	T	T	T	T	T	T	T	T	T	T	F
T	T	F	T	T	T	F	F	T	T	F	T
T	F	T	F	T	F	T	T	T	F	T	F
T	F	F	F	T	F	F	T	T	F	F	T
F	T	T	T	F	T	T	T	F	T	T	F
F	T	F	T	F	T	F	F	F	T	F	F
F	F	T	T	F	F	T	T	F	F	T	F
F	F	F	T	F	F	F	T	F	F	F	F

Now it is easy to check. Since there is no row where all three wffs have a T under the stopper, this array is a proof of unsatisfiability in the sense defined in 4.2.1. above.

We may also regard the resulting array of top-row-identical wffs as a proof that the original array is unsatisfiable, on the grounds that it is easy enough to check whether a larger truth table is an expansion of a smaller one. Of course, that does not mean that the original array, which is not top-row-identical, is *itself* a proof of its own satisfiability.

At this point, a student of the *Tractatus* might want to appeal to the early Wittgenstein's notion of what is essential in a proposition (or more generally, in a symbol):

> 3.341 The essential in a proposition is therefore that which is common to all propositions which can express the same sense.
>
> And in the same way in general the essential in a symbol is that which all symbols which can fulfil the same purpose have in common.

Thus, the original array, regarded as an array of propositions (propositions in the sense of the *Tractatus*), may be said, following Wittgenstein, to be *essentially the same* as the expanded top-row-identical array. But we must be careful here not to kid ourselves. Insofar as formal proofs are (or at least, constitutively involve) *particular typographical objects*, only the expanded version should count as a proof, and this must be regarded as a distinct object from the unexpanded version.

V. CONCLUSION

Wittgenstein's suggestion that truth tables may be seen as propositional signs has been developed. The result is a precisely defined language which gives rise to a distinctive consequence relation and a distinctive domain of facts about properties of logical interest. We saw that single wffs written in this notation can serve as their own proofs with respect to properties of logical interest, as can some groups of wffs. I argued that not all finite groups of wffs enjoy this self-proving property, on the grounds that proof-checking ought to be easy. I then gave two alternative responses to this problem: the system of row tree proofs, shown to be sound and complete for its intended domain, and the rewriting strategy.

In my view, Wittgenstein's attitude toward truth tables as propositional signs has been both vindicated and tempered. I detect a utopian element in Wittgenstein's logical thought, regarding what we would get if we had an optimal symbolism; as though, if we just hit upon the right way of writing things, we wouldn't need to muck around manipulating symbols, and could instead just look at what we had written. The present investigation suggests that this is in an important sense not so and may contribute to our understanding of why not.

It may be replied on Wittgenstein's behalf that, provided we have some fixed set of atoms in our language and require that each wff contains all the atoms in some particular order, we can retain the self-proving property. While this is true, there are counter-replies that can be made which support our "no free lunch" stance. Firstly, if the number of atoms is not small, it will be very difficult and cumbersome to write anything. Having 20 atoms in play – and one can easily imagine domains of discourse where there are 20 or more independently varying matters of interest – would make wffs over a million lines long (and of course this number gets further out of hand very quickly). So yes, maybe in this way one can say that testing for metalogical properties remains easy *given* some wffs, but then it may become very difficult to write anything in the first place. This just moves the work somewhere else. (I don't say this is insignificant: one could imagine a strategic game-like scenario where one player has agreed to check groups of formulas, another has agreed to provide formulas containing information, and the first player gets to choose the notation. The present considerations suggest that it might be a good idea for the first player to choose this sort of notation.) Secondly, we may want to leave it open how many atoms we are going to use. We might want to start small and expand when needed, without moving to a distinct language. This benefit appears to be incompatible with preserving the self-proving property for arbitrary groups of formulas.

Finally, from a technical point of view, it may be interesting to consider various extensions of the language of TPL (for instance in the direction of modal or predicate logic), and to consider further from a computational complexity perspective the problem of forming a semantic conjunction of two or more truth tables.[9]

Faulty of Arts
The University of Melbourne
Room 659, Level 6 North Wing, Building 148, Arts West
Victoria 3010, Australia
E-mail: tristan.grotvedt@unimelb.edu.au

NOTES

[1] In some presentations of propositional logic, connectives are not defined in terms of truth tables. Rather, some are taken as primitive (i.e., not defined at all), and others defined in terms of the primitive ones. However, in other presentations of propositional logic, such as that found in Chapter 3 of Smith (2012) (a popular textbook), truth tables are used to define the connectives.

[2] See Wittgenstein (1922), §4.442. For some elaborations of this idea in Wittgenstein's lectures in the early 1930's, see Wittgenstein (2016), p. 60, p. 114, and p. 130.

[3] It seems likely to the present author that some consideration of this sort lies behind Wittgenstein's use of the letters 'a' and 'b' rather than 'T' and 'F' in the related 'ab-Notation' for propositional logic which he devised, although this later became 'TF-notation' in the *Tractatus*. [For background on ab-Notation see Wittgenstein (1961), Potter (2009)].

[4] For present purposes it is convenient to take arrays as primitive, but note that an n-by-m array may be treated as an n-tuple of m-tuples, which in turn may be treated as a certain kind of set.

[5] This is significant from the point of view of Wittgenstein's motivation for thinking of truth tables as propositional signs. For Wittgenstein one desideratum of a logical notation is that, as Landini (2011), p. 44 puts it, 'all and only logical equivalents have the same expression'. For instance, in the early *Notes on Logic*, Third MS, we find the remark: 'If p = not-not-p etc., this shows that the traditional method of symbolism is wrong, since it allows a plurality of symbols with the same sense; and thence it follows that, in analyzing such propositions,

we must not be guided by Russell's method of symbolizing.' (These notes appear in Wittgenstein (1961) and also in Potter (2009) (the main text of which is an extended commentary on them).)

[6] If the notion of formal proof is being introduced only with respect to a particular form of proof, then these two requirements may suffice. For example, in the context of axiom systems, where checking is typically feasible. (However, note that the feasibility of checking a putative axiomatic proof does depend on what axioms are allowed. One sometimes sees, in specifications of axiom systems for modal logic, words to the effect that all instances of propositionally tautologous forms count as axioms. For the reasons discussed above, the status of "proofs" in such axiom systems *as formal proofs* is on shaky ground. Thanks to N.J.J. Smith for discussion on this point.)

[7] See Cook and Reckhow (1973). Following Cook and Reckhow (1973), Beame and Pitassi precisify 'easy to check' in terms of there being a proof-checking algorithm that runs in polynomial time. (Note that this is a point about proof-*checking* — not proof construction.) This is, I think, a reasonable necessary condition on formal proofs. However, I do not think this condition is sufficient. Many things doable in polynomial or even linear time are still hard to do from an intuitive point of view, and proofs ought to be easy to check. (Thanks to N.J.J. Smith for discussion on this point.)

[8] Many thanks to an anonymous referee for this journal for suggesting this strategy.

[9] Many thanks to Fengning Yang for reading an early draft of this paper and providing several comments which led to improvements.

References

BEAME, P. & PITASSI, T. (2001), 'Propositional Proof Complexity: Past, Present, and Future'; in Paun, G., Rosenberg, G. & Salomaa, A. (eds.), *Current Trends in Theoretical Computer Science: Entering the 21st Century*, Singapore, *World Scientific Publishing Co*., pp. 42-70.

COOK, S.A. & RECKHOW, R.A. (1973), 'Time-Bounded Random Access Machines'; *Journal of Computer Systems Science*, vol. VII, pp. 354-375.

GIAMMARRESI, D., RESTIVO, A. (1997), 'Two-Dimensional Languages', in Rozenberg G. and Salomaa A. (eds.) *Handbook of Formal Languages*. Berlin, Springer, pp. 215-267.

HILBERT, D. (1918), 'Axiomatisches Denken', *Mathematische Annalen*, vol. LXXVIII, pp. 405–415. (Lecture given at the Swiss Society of Mathematicians, 11 September 1917.) English translation 'Axiomatic Thought'; in Ewald, W.B. (ed.) (1996), *From Kant to Hilbert. A Source Book in the Foundations of Mathematics,* vol. II, Oxford, Oxford University Press, pp. 1105-1115.

LANDINI, G. (2011), 'Wittgenstein Reads Russell'; in McGinn, M. & Oskari Kuusela, O. (eds.), *The Oxford Handbook of Wittgenstein*, Oxford, Oxford University Press.

POTTER, M. (2009), *Wittgenstein's Notes on Logic*; Oxford, Oxford University Press.

SMITH, N.J.J. (2012), *Logic: The Laws of Truth*; Princeton, Princeton University Press.

WITTGENSTEIN, L. (1922), *Tractatus Logico-Philosophicus*; trans. C. K. Ogden, C.K, London, Routledge and Kegan Paul Ltd.

— (1961), *Notebooks 1914–1916*; von Wright, G.H. and Anscombe, G.E.M. (eds.), Oxford, Blackwell.

— (2016), *Lectures, Cambridge 1930–1933: From the Notes of G. E. Moore*; Stern, D.G., Rogers, B. & Citron, G. (eds.), Cambridge, Cambridge University Press.