7.1 Truth tables from the perpective of evidence semantics

The main aspect of boolean, or truth table, logics is the fundamental assumption that the law of excluded middle holds for every propositional variable and thus, as we have shown inductively, for every formula. Thus a proof of a formula X implicitly assumes $p \vee \neg p$ for every propositional variable p that occurs in the formula X. For instance, proving the formula $(P \wedge Q) \Rightarrow P$ actually means proving $((P \vee \neg P) \wedge (Q \vee \neg Q)) \Rightarrow ((P \wedge Q) \Rightarrow P)$.

Truth tables are a popular method for establishing the validity of a formula X. In a table we use one column per variable occurring in X and the formula X itself. If X is a complx formula we also include columns for the subformulas of X. Afterwards we write all possible boolean valuations into the rows of this table. In other words, we simply write down all possible combinations of assignments `t` and `f` to the variables of the formula and then compute the boolean value all of the formulas in the table. If every row shows a `t` in the column for X, then we know that X is true under all possible valuations and thus valid.

Example 7.1 Here is a truth table proof for the validity of the formula $(P \wedge Q) \Rightarrow P$.

P	Q	$P \wedge Q$	$(P \wedge Q) \Rightarrow P$
t	t	t	t
t	f	f	t
f	t	f	t
f	f	f	t

After entering the assignments for P and Q we calculate the values for $P \wedge Q$ (`t` if one of the two values is `t` and `f` otherwise) and from that the values for $(P \wedge Q) \Rightarrow P$ (`t` if the value for P is `t` or the value for $P \wedge Q$ is `f`). Since all the entries in the final column are `t` $(P \wedge Q) \Rightarrow P$ is valid. □

By providing one row for for each possible combination of assignments `t` and `f` to the variables of a formula a truth table proof makes explicit that every formula is proved under the assumption $p \vee \neg p$ for every propositional variable p that occurs in it. From the perspective of evidence, the assignment `t` in a column for a variable p indicates the assumption that we know p and the assignment `f` indicates that we have evidence for $\neg p$. In the same way, an entry `t` in the column for a formula X (or one of its subformulas) means that we know X or, to be precise, that we know that there must be evidence for X, and an entry `f` means that there must be evidence for $\neg X$.

So in a sense, a truth table is a shorthand notation for a more complex *evidence table*, which states that if we have evidence for all the assumptions $p \vee \neg p$, where p is a variable ocurring in X then we are able to *decide* X, i.e. provide evidence for $X \vee \neg X$. Each row of the table describes the possible combinations of evidences that we may have for the assumptions and the formula X is valid if the evidence provided for $X \vee \neg X$ always points to X, i.e. the left disjunct.

Example 7.2 The simplest evidence tables are tables for conjunction, disjunction, implication, and negation of two propositional variables P and Q. To construct the evidence table for such a formula, we provide columns for the assumptions $P \vee \neg P$ and $Q \vee \neg Q$ andand columns for the evidences for the target formulas $(P \wedge Q) \vee \neg(P \wedge Q)$, $(P \vee Q) \vee \neg(P \vee Q)$, $(P \Rightarrow Q) \vee \neg(P \Rightarrow Q)$, or $(\neg P) \vee \neg(\neg P)$, respectively. An evidence for $P \vee \neg P$ will either have the form $\texttt{inl}(p)$, where p is evidence for P, i.e. an element of $[P]$, or the form $\texttt{inr}(n_p)$, where n_p is evidence for $\neg P$, i.e. a function in $[P] \to \{\}$). Similarly there are two possible evidences $\texttt{inl}(q)$ and $\texttt{inr}(n_q)$ for $Q \vee \neg Q$, which gives us four possible combinations of evidences.

$P \vee \neg P$	$Q \vee \neg Q$	$(P \wedge Q) \vee \neg(P \wedge Q)$	$(P \vee Q) \vee \neg(P \vee Q)$	$(P \Rightarrow Q) \vee \neg(P \Rightarrow Q)$	$(\neg P) \vee \neg(\neg P)$
$\texttt{inl}(p)$	$\texttt{inl}(q)$				
$\texttt{inl}(p)$	$\texttt{inr}(n_q)$				
$\texttt{inr}(n_p)$	$\texttt{inl}(q)$				
$\texttt{inr}(n_p)$	$\texttt{inr}(n_q)$				

Let us analyze how to fill in the entries for each of the target formulas.

$P \wedge Q$: We know that $P \wedge Q$ only has evidence if both P and Q have and that the evidence will have the form (p, q) where p:$[P]$ and q:$[Q]$. Since this fits our assumptions for the first row, the entry will be $\texttt{inl}(p, q)$ in this case. In all other cases there will be no evidence for $P \wedge Q$, so we have to construct evidence of the form $\texttt{inr}(f)$, where f will be a function from $[P] \times [Q]$ to $\{\}$. In order to build this function assume that we have an input x:$[P] \times [Q]$ and show how to map this into an element of $\{\}$, using the evidences n_p and n_q. In row 2 we apply $n_q : [Q] \to \{\}$ to x_2:$[Q]$ and in row 3 we apply $n_p : [P] \to \{\}$ to x_1:$[P]$. In row 4 we may use either $n_p(x_1)$ or $n_q(x_2)$.

$P \vee \neg P$	$Q \vee \neg Q$	$(P \wedge Q) \vee \neg(P \wedge Q)$
$\texttt{inl}(p)$	$\texttt{inl}(q)$	$\texttt{inl}(p, q)$
$\texttt{inl}(p)$	$\texttt{inr}(n_q)$	$\texttt{inr}(\lambda x. n_q(x_2))$
$\texttt{inr}(n_p)$	$\texttt{inl}(q)$	$\texttt{inr}(\lambda x. n_p(x_1))$
$\texttt{inr}(n_p)$	$\texttt{inr}(n_q)$	$\texttt{inr}(\lambda x. n_p(x_1))$ $\texttt{inr}(\lambda x. n_q(x_2))$

$P \vee Q$: We know that $P \vee Q$ only has evidence if either P and Q have and that the evidence will have the form $\texttt{inl}(p)$ where p:$[P]$ to indicate that we prove the left disjunct of $P \vee Q$, or the form $\texttt{inr}(q)$ where q:$[Q]$ to indicate that we prove the right disjunct. Thus in the second and third row, the corresponding entries will be $\texttt{inl(inl}(p))$ and $\texttt{inl(inr}(q))$ to indicate that we show the left disjunct of $(P \vee Q) \vee \neg(P \vee Q)$. In the first row we may use both $\texttt{inl(inl}(p))$ and $\texttt{inl(inr}(q))$ as evidences.

Only in the case that we have evidence for $\neg P$ and for $\neg Q$ there can be no evidence for $P \vee Q$ and we have to construct evidence for its negation. This evidence has to be a function that takes an element of $[P] + [Q]$ as input and constructs an element of $\{\}$. Since the input element is either $\texttt{inl}(p)$ for some p:$[P]$ or $\texttt{inr}(q)$ for some q:$[Q]$, we may apply n_p to p or n_q to q for this purpose. Thus $\lambda x.\, \texttt{case}\ x\ \texttt{of}\ \texttt{inl}(p) \to n_p(p)\ |\ \texttt{inr}(q) \to n_q(q)$ is the evidence for $\neg(P \vee Q)$.

$P \vee \neg P$	$Q \vee \neg Q$	$(P \vee Q) \vee \neg(P \vee Q)$
$\texttt{inl}(p)$	$\texttt{inl}(q)$	$\texttt{inl(inl}(p))$, $\texttt{inl(inr}(q))$
$\texttt{inl}(p)$	$\texttt{inr}(n_q)$	$\texttt{inl(inl}(p))$
$\texttt{inr}(n_p)$	$\texttt{inl}(q)$	$\texttt{inl(inr}(q))$
$\texttt{inr}(n_p)$	$\texttt{inr}(n_q)$	$\texttt{inr}(\lambda x.\, \texttt{case}\ x\ \texttt{of}\ \texttt{inl}(p) \to n_p(p) \mid \texttt{inr}(q) \to n_q(q))$

$P \Rightarrow Q$: We know that $P \Rightarrow Q$ has evidence if there is a way to construct evidence for Q from evidence for P. This is obviously the case if we already have an evidence q for Q: the evidence would simply be $\lambda x.q$, i.e. a function that takes an input $x : [P]$ and returns q.If we have evidence n_p for $\neg P$ then applying n_p to an element $p{:}P$ would give us an element of $\{\}$ and, following our discussion in lecture 4, we may use the special function $\mathtt{any} : \{\} \rightarrow [Q]$ to build the evidence for Q from that. The evidence for $P \Rightarrow Q$ would thus be $\lambda x.\mathtt{any}(n_p(x))$.

Only in the case that we have evidence for P and for $\neg Q$ (row 2) there can be no evidence for $P \Rightarrow Q$ and we have to construct evidence for its negation. This evidence has to be a function that takes as input a function $f : [P] \Rightarrow [Q]$ and constructs an element of $\{\}$. For the latter, we may apply n_q to evidence for Q, which we get by applying f to the evidence p for P. Thus the evidence for $\neg(P \Rightarrow Q)$ will be $\lambda f.n_q(f(p))$.

If we add the appropriate `inl` and `inr` functions to indicate for which of the two disjuncts we have constructed the evidence we get the following table for $(P \Rightarrow Q) \vee \neg(P \Rightarrow Q)$.

$P \vee \neg P$	$Q \vee \neg Q$	$(P \Rightarrow Q) \vee \neg(P \Rightarrow Q)$
$\mathtt{inl}(p)$	$\mathtt{inl}(q)$	$\mathtt{inl}(\lambda x.q)$
$\mathtt{inl}(p)$	$\mathtt{inr}(n_q)$	$\mathtt{inr}(\lambda f.n_q(f(p)))$
$\mathtt{inr}(n_p)$	$\mathtt{inl}(q)$	$\mathtt{inl}(\lambda x.q)$
$\mathtt{inr}(n_p)$	$\mathtt{inr}(n_q)$	$\mathtt{inl}(\lambda x.\mathtt{any}(n_p(x)))$

$\neg P$: The evidence table for negation only needs two rows stating that we either assume to have evidence p for P or n_p for $\neg P$. In the second case n_p is already the evidence for the left disjunct of $(\neg P) \vee \neg(\neg P)$. In the first, we need to describe a function that takes as input a function $f : [P] \Rightarrow \{\}$ and constructs an element of $\{\}$. Since we assume to have the element $p{:}[P]$ we only have to apply f to p.

$P \vee \neg P$	$(\neg P) \vee \neg(\neg P)$
$\mathtt{inl}(p)$	$\mathtt{inr}(\lambda f.f(p))$
$\mathtt{inr}(n_p)$	$\mathtt{inl}(n_p)$

□

If we compare the evidence tables in the above example with the truth tables for conjunction, disjunction, implication, and negation, we can make an interesting observation.

P	Q	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$\neg P$
t	t	t	t	t	f
t	f	f	t	f	f
f	t	f	t	t	t
f	f	f	f	t	t

Each entry of the form `inl(...)` in the evidence tables corresponds to an entry `t` in the truth table and each entry of the form `inr(...)` corresponds to an `f`. In a sense `t` and `f` are abbreviations for evidence terms that reduce the amount of information provided. An entry `t` in the column of a formula X only tells us that X holds but does not tell us why this is the case. Similarly, an entry `f` tells us that the negation of X holds but does not tell us why. Thus truth tables are condensed forms of evidence tables that only present the result of a decision (i.e. whether X or $\neg X$ holds) and drop all the information about the *why*. To construct a truth table from an evidence table we only look at the outer structure of the evidence, convert each `inl` into a `t` and each `inr` into an `f` and drop the rest.

How do we build evidence tables for more complex formulas? We could proceed as above and use semantical arguments to construct the evidence for each case directly. This approach usually leads to simple evidences but becomes increasingly difficult as formulas grow in size. An alternative is to follow the truth-table approach and compose the above evidence tables for conjunction, disjunction, implication, and negation to build evidences for all the subformulas of a target formula. This will enable us to build tables for arbitrary complicated formulas but may leqad to unnecessarily complicated evidences.

Example 7.3 To construct the evidence table for the formula $(P \wedge Q) \Rightarrow P$ in a compositional fashion we provide columns for the assumptions $P \vee \neg P$ and $Q \vee \neg Q$ and columns for the evidences for $(P \wedge Q) \vee \neg(P \wedge Q)$ and $((P \wedge Q) \Rightarrow P) \vee \neg((P \wedge Q) \Rightarrow P)$. We already know the evidences for $(P \wedge Q) \vee \neg(P \wedge Q)$ and fill the table accordingly.

$P \vee \neg P$	$Q \vee \neg Q$	$(P \wedge Q) \vee \neg(P \wedge Q)$	$((P \wedge Q) \Rightarrow P) \vee \neg((P \wedge Q) \Rightarrow P)$
$\texttt{inl}(p)$	$\texttt{inl}(q)$	$\texttt{inl}(p, q)$	
$\texttt{inl}(p)$	$\texttt{inr}(n_q)$	$\texttt{inr}(\lambda x.n_q(x_2))$	
$\texttt{inr}(n_p)$	$\texttt{inl}(q)$	$\texttt{inr}(\lambda x.n_p(x_1))$	
$\texttt{inr}(n_p)$	$\texttt{inr}(n_q)$	$\texttt{inr}(\lambda x.n_p(x_1))$ $\texttt{inr}(\lambda x.n_q(x_2))$	

We then apply the evidence table for implication to the evidence in the third and first column of our table. That is we use the evidence terms for $(P \wedge Q) \vee \neg(P \wedge Q)$ and $P \vee \neg P$ as inputs for the implication table. In the first row, we build a function of the form $\texttt{inl}(\lambda x.p)$, where p is the evidence in the first column and x is an arbitary element in $[P] \times [Q]$, so the evidence we need is $\texttt{inl}(\lambda x.p)$. Row two matches the third row of the implication table and also gives us $\texttt{inl}(\lambda x.p)$. Row three corresponds to row four of the implication table yields $\texttt{inl}(\lambda x.\texttt{any}((\lambda x.n_q(x_1))(x)))$, which we reduce to $\texttt{inl}(\lambda x.\texttt{any}(n_p(x_1)))$. We get the same result in row four. Alternatively we could also use $\texttt{inl}(\lambda x.\texttt{any}(n_q(x_2)))$.

$P \vee \neg P$	$Q \vee \neg Q$	$(P \wedge Q) \vee \neg(P \wedge Q)$	$((P \wedge Q) \Rightarrow P) \vee \neg((P \wedge Q) \Rightarrow P)$
$\texttt{inl}(p)$	$\texttt{inl}(q)$	$\texttt{inl}(p, q)$	$\texttt{inl}(\lambda x.p)$
$\texttt{inl}(p)$	$\texttt{inr}(n_q)$	$\texttt{inr}(\lambda x.n_q(x_2))$	$\texttt{inl}(\lambda x.p)$
$\texttt{inr}(n_p)$	$\texttt{inl}(q)$	$\texttt{inr}(\lambda x.n_p(x_1))$	$\texttt{inl}(\lambda x.\texttt{any}(n_p(x_1)))$
$\texttt{inr}(n_p)$	$\texttt{inr}(n_q)$	$\texttt{inr}(\lambda x.n_p(x_1))$ $\texttt{inr}(\lambda x.n_q(x_2))$	$\texttt{inl}(\lambda x.\texttt{any}(n_p(x_1)))$ $\texttt{inl}(\lambda x.\texttt{any}(n_q(x_2)))$

Since we get evidence for the left disjunct in all cases we know that $(P \wedge Q) \Rightarrow P$ is valid under the assumptions $P \vee \neg P$ and $Q \vee \neg Q$. □

In contrast to truth tables, evidence tables do not only tell us whether a formula is valid. They are also a tabular description of evidence algorithms, providing specific evidences for all possible combinations of decisions. To extract this algorithm from the tabke, one simply collects all these cases into a case analysis of all the "decision variables" involved[1] and creates a term of the form

[1]To separate the evidence variables related to decisions $P \vee \neg P$, $Q \vee \neg Q$, etc. we denote them by $d_1, d_2, d_3, \ldots$. These variables will be acccessed only through case analysis and should not be confused with components of a pair.

$$
\begin{array}{l}
\lambda d_1.\lambda d_2....\lambda d_n \\
\quad \texttt{case}\ d_1\ \texttt{of}\ \texttt{inl}(p) \to \\
\qquad\qquad \texttt{case}\ d_2\ \texttt{of}\ \texttt{inl}(q) \to \ \ldots \\
\qquad\qquad\qquad\quad \mid\ \texttt{inr}(n_q) \to \ \ldots \\
\qquad\quad \mid\ \texttt{inr}(n_p) \to \\
\qquad\qquad \texttt{case}\ d_2\ \texttt{of}\ \texttt{inl}(q) \to \ \ldots \\
\qquad\qquad\qquad\quad \mid\ \texttt{inr}(n_q) \to \ \ldots
\end{array}
$$

For instance, the extracted evidence for $((P \wedge Q) \Rightarrow P) \vee \neg((P \wedge Q) \Rightarrow P)$ extracted would be

$$
\begin{array}{l}
\lambda d_1.\lambda d_2. \\
\quad \texttt{case}\ d_1\ \texttt{of}\ \texttt{inl}(p) \to \\
\qquad\qquad \texttt{case}\ d_2\ \texttt{of}\ \texttt{inl}(q) \to \ \texttt{inl}(\lambda x.p) \\
\qquad\qquad\qquad\quad \mid\ \texttt{inr}(n_q) \to \ \texttt{inl}(\lambda x.p) \\
\qquad\quad \mid\ \texttt{inr}(n_p) \to \\
\qquad\qquad \texttt{case}\ d_2\ \texttt{of}\ \texttt{inl}(q) \to \ \texttt{inl}(\lambda x.\mathrm{any}(n_p(x_1))) \\
\qquad\qquad\qquad\quad \mid\ \texttt{inr}(n_q) \to \ \texttt{inl}(\lambda x.\mathrm{any}(n_p(x_1)))
\end{array}
$$

This term can obviously be simplified, as the variable d_2, which decides $Q \vee Q$, does not occur in either of the cases. After joining the identical cases we get

$$
\begin{array}{l}
\lambda d_1.\lambda d_2.\ \texttt{case}\ d_1\ \texttt{of}\ \texttt{inl}(p) \to \ \texttt{inl}(\lambda x.p) \\
\qquad\qquad\qquad\quad \mid\ \texttt{inr}(n_p) \to \ \texttt{inl}(\lambda x.\mathrm{any}(n_p(x_1)))
\end{array}
$$

This algorithm shows that only one of the two possible decisions is needed to establish the validity of $(P \wedge Q) \Rightarrow P$[2] . This means that the space of decisions that need to be made has only *dimension 1* instead of being two-dimensional. The following example shows that its dimension can actually be reduced to 0.

Example 7.4 In the specific example of $(P \wedge Q) \Rightarrow P$ it is much simpler to construct the evidence table using semantical arguments instead of composing basic evidence tables. We know that the evidence for $(P \wedge Q) \Rightarrow P$ is a function that takes as input an evidence for $P \wedge Q$, i.e. an element x of $[P] \times [Q]$, and returns an evidence for P. For this purpose we may simply use the first component of x and the evidence in all four cases is $\texttt{inl}(\lambda x.x_1)$.

$P \vee \neg P$	$Q \vee \neg Q$	$((P \wedge Q) \Rightarrow P) \vee \neg((P \wedge Q) \Rightarrow P)$
$\texttt{inl}(p)$	$\texttt{inl}(q)$	$\texttt{inl}(\lambda x.x_1)$
$\texttt{inl}(p)$	$\texttt{inr}(n_q)$	$\texttt{inl}(\lambda x.x_1)$
$\texttt{inr}(n_p)$	$\texttt{inl}(q)$	$\texttt{inl}(\lambda x.x_1)$
$\texttt{inr}(n_p)$	$\texttt{inr}(n_q)$	$\texttt{inl}(\lambda x.x_1)$

This shows that the decision variables are not required for the construction of the evidence table at all. The algorithm extracted from this table is $\lambda d_1.\lambda d_2.\ \texttt{inl}(\lambda x.x_1)$, which means that the decision space for the formula $(P \wedge Q) \Rightarrow P$ has the dimension 0. □

Evidence tables also reveal the subtle differences between formulas that cannot be distinguished using truth tables.

[2]We could simplify the algorithm further by moving the inl out, which shows the validity of the formula. As a consequence the variable x in the inner λ-expression always refers to an element of type $[P] \times [Q]$, which allows us to move the λ-abstraction out as well: $\lambda d_1.\lambda d_2.\ \texttt{inl}(\lambda x.\ \texttt{case}\ d_1\ \texttt{of}\ \texttt{inl}(p) \to\ p \mid \texttt{inr}(n_p) \to\ \mathrm{any}(n_p(x_1)))$. Further optimizations are only possible using semantical arguments as in example 7.4.

Example 7.5 The formulas $P \Rightarrow Q$ and $\neg P \vee Q$ are considered equal from the perspective of truth tables, since the truth table entries are identical in all four cases.

P	Q	$\neg P$	$P \Rightarrow Q$	$\neg P \vee Q$
t	t	f	t	t
t	f	f	f	f
f	t	t	t	t
f	f	t	t	t

If we look at the evidence tables instead we see that, while the discision itself is the same in all four rows, the specific evidences are very different. The evidence for $P \Rightarrow Q$ is always a function while the evidence for $\neg P \vee Q$ is an injection.

$P \vee \neg P$	$Q \vee \neg Q$	$(P \Rightarrow Q) \vee \neg(P \Rightarrow Q)$	$(\neg P \vee Q) \vee \neg(\neg P \vee Q)$
$\texttt{inl}(p)$	$\texttt{inl}(q)$	$\texttt{inl}(\lambda x.q)$	$\texttt{inl}(\texttt{inr}(q))$
$\texttt{inl}(p)$	$\texttt{inr}(n_q)$	$\texttt{inr}(\lambda f.n_q(f(p)))$	$\texttt{inr}(\lambda x.\,\texttt{case}\ x\ \texttt{of}\ \ \texttt{inl}(f) \to f(p) \mid \texttt{inr}(q) \to n_q(q))$
$\texttt{inr}(n_p)$	$\texttt{inl}(q)$	$\texttt{inl}(\lambda x.q)$	$\texttt{inl}(\texttt{inl}(n_p))$
$\texttt{inr}(n_p)$	$\texttt{inr}(n_q)$	$\texttt{inl}(\lambda x.\texttt{any}(n_p(x)))$	$\texttt{inl}(\texttt{inl}(n_p))$

□