| | |
|---|---|
| $\dfrac{\sigma\,\mathbb{T}\,\varphi \land \psi}{\begin{array}{c}\sigma\,\mathbb{T}\,\varphi \\ \sigma\,\mathbb{T}\,\psi\end{array}}\land\mathbb{T}$ | $\dfrac{\sigma\,\mathbb{F}\,\varphi \land \psi}{\sigma\,\mathbb{F}\,\varphi \quad \mid \quad \sigma\,\mathbb{F}\,\psi}\land\mathbb{F}$ |
| $\dfrac{\sigma\,\mathbb{T}\,\varphi \lor \psi}{\sigma\,\mathbb{T}\,\varphi \quad \mid \quad \sigma\,\mathbb{T}\,\psi}\lor\mathbb{T}$ | $\dfrac{\sigma\,\mathbb{F}\,\varphi \lor \psi}{\begin{array}{c}\sigma\,\mathbb{F}\,\varphi \\ \sigma\,\mathbb{F}\,\psi\end{array}}\lor\mathbb{F}$ |

Table 1: Prefixed tableau rules for $\land$ and $\lor$

int:tab:rul:
tab:prop-rules

## tab.1 Rules for Intuitionistic Logic

int:tab:rul:
sec

The rules for the connectives $\land$ and $\lor$ are the same as for regular propositional signed tableaux, just with prefixes added. In each case, the rule applied to a signed formula $\sigma\,S\,\varphi$ produces new formulas that are also prefixed by $\sigma$. This should be intuitively clear: e.g., if $\varphi \land \psi$ is true at (a world named by) $\sigma$, then $\varphi$ and $\psi$ are true at $\sigma$ (and not at any other world). We collect the rules for $\land$ and $\lor$ in Table 1.

The closure condition is similar to that for ordinary tableaux, although we require that not just the formulas, but also that the prefixes must match. In fact, we can be somewhat more liberal: Since in intuitionistic models, formulas, once true, remain true, it is impossible that $\varphi$ is true at $\sigma$ but false at any accessible prefix $\sigma.*$. So a branch is closed if it contains both

$$\sigma\,\mathbb{T}\,\varphi \quad \text{and} \quad \sigma.*\,\mathbb{F}\,\varphi$$

for some prefix $\sigma$ and formula $\varphi$. Note that if the signs are reversed, i.e., if it contains

$$\sigma\,\mathbb{F}\,\varphi \quad \text{and} \quad \sigma.*\,\mathbb{T}\,\varphi$$

the branch is closed only if $*$ is the empty sequence.

In addition, a branch is closed if it contains $\sigma\,\mathbb{T}\,\bot$.

The rules for setting up assumptions is also as for ordinary tableaux, except that for assumptions we always use the prefix 1. (It does not matter which prefix we use, as long as it's the same for all assumptions.) So, e.g., we say that

$$\psi_1, \dots, \psi_n \vdash \varphi$$

iff there is a closed tableau for the assumptions

$$1\,\mathbb{T}\,\psi_1, \dots, 1\,\mathbb{T}\,\psi_n, 1\,\mathbb{F}\,\varphi.$$

For the conditional $\to$, the rules differ from the classical and modal cases. The $\mathbb{T}{\to}$ rule extends a branch containing $\sigma\,\mathbb{T}\,\varphi \to \psi$ by $\sigma.*\,\mathbb{T}\,\varphi$ and $\sigma.*\,\mathbb{F}\,\psi$ on two different branches. It can only be applied for a prefix $\sigma.*$ which *already*

| | |
|---|---|
| $\dfrac{\sigma\,\mathbb{T}\,\neg\varphi}{\sigma.{*}\,\mathbb{F}\,\varphi}\;\neg\mathbb{T}$ <br> $\sigma.{*}$ is used | $\dfrac{\sigma\,\mathbb{F}\,\neg\varphi}{\sigma.n\,\mathbb{T}\,\varphi}\;\neg\mathbb{F}$ <br> $\sigma.n$ is new |
| $\dfrac{\sigma\,\mathbb{T}\,\varphi\to\psi}{\sigma.{*}\,\mathbb{F}\,\varphi \quad \mid \quad \sigma.{*}\,\mathbb{T}\,\psi}\;{\to}\mathbb{T}$ <br> $\sigma.{*}$ is used | $\dfrac{\sigma\,\mathbb{F}\,\varphi\to\psi}{\begin{array}{c}\sigma.n\,\mathbb{T}\,\varphi\\ \sigma.n\,\mathbb{F}\,\psi\end{array}}\;{\to}\mathbb{F}$ <br> $\sigma.n$ is new |

Table 2: Prefixed tableau rules for $\neg$ and $\to$

int:tab:rul:
tab:rules-lif-lnot

occurs on the branch in which it is applied. Let's call such a prefix "used" (on the branch). (Since $\sigma.{*}$ includes $\sigma$ itself, the rule can always be applied by adding the prefixed signed formulas $\sigma\,\mathbb{T}\,\varphi$ and $\sigma\,\mathbb{F}\,\psi$ on separate branches.)

The $\mathbb{F}{\to}$ rule extends a branch containing $\sigma\,\mathbb{F}\,\varphi\to\psi$ by both $\sigma.n\,\mathbb{T}\,\varphi$ and $\sigma.n\,\mathbb{F}\,\psi$ on the same branch, with $\sigma.n$ a prefix new to the branch.

The rules for $\neg$ are defined analogously (using the definition of $\neg\varphi$ as $\varphi\to\bot$).

The rules are given in Table 2.

## Photo Credits

## Bibliography