Sentential Tree Logic

Logic I
Benjamin Brast-McKie
September 25, 2023

Constructing the Root

Previously: $\Gamma \vDash \varphi$ if and only if $\Gamma, \neg\varphi \vDash \bot$.

Proof: To show $\Gamma, \neg\varphi \vDash \bot$, we will show $\Gamma, \neg\varphi \vdash \bot$.

Resolution Rules

Root: A *root* is any finite sequence of SL sentences.

Tree: An SL *tree* consists of a root followed by any number of applications of the resolution rules given below:

Conjunction

$$\begin{array}{c} \varphi \wedge \psi \\ \varphi \\ \psi \end{array} \qquad \begin{array}{c} \neg(\varphi \wedge \psi) \\ \diagup \quad \diagdown \\ \neg\varphi \qquad \neg\psi \end{array}$$

The Material Conditional

$$\begin{array}{c} \varphi \supset \psi \\ \diagup \quad \diagdown \\ \neg\varphi \qquad \psi \end{array} \qquad \begin{array}{c} \neg(\varphi \supset \psi) \\ \varphi \\ \neg\psi \end{array}$$

Disjunction

$$\begin{array}{c} \varphi \vee \psi \\ \diagup \quad \diagdown \\ \varphi \qquad \psi \end{array} \qquad \begin{array}{c} \neg(\varphi \vee \psi) \\ \neg\varphi \\ \neg\psi \end{array}$$

The Biconditional

$$\begin{array}{c} \varphi \equiv \psi \\ \diagup \quad \diagdown \\ \begin{array}{cc} \varphi & \neg\varphi \\ \psi & \neg\psi \end{array} \end{array} \qquad \begin{array}{c} \varphi \equiv \psi \\ \diagup \quad \diagdown \\ \begin{array}{cc} \varphi & \neg\varphi \\ \neg\psi & \psi \end{array} \end{array}$$

Double Negation

$$\begin{array}{c} \neg\neg\varphi \\ \varphi \end{array}$$

Child: A *child* of φ in an SL tree is any sentence ψ immediately following φ.

Leaf: A *leaf* in an SL tree is any sentence which does not have a child.

Branch: A *branch* in an SL tree is any sequence of sentences beginning with the root of the tree and ending with a leaf of the tree where every sentence in the sequence besides the first is a child of its predecessor.

Note: Officially, an SL tree is an *ordered dyadic tree* of SL sentences where every sentence in the tree either belongs to the root, or results from resolving one of its ancestors.

Closure and Completion

Branch Closure: A branch in an SL tree is *closed* just in case it includes φ and $\neg\varphi$ for some SL sentence φ, and *open* otherwise.

Tree Closure: A tree is *closed* just in case every branch is closed, and *open* otherwise.

Resolvable: A sentence is *resolvable* in a branch just in case it has a resolution rule and the branch is open.

Resolved: A sentence is *resolved* in a branch just in case the resolution rule for that sentence has been applied in that branch.

Branch Completion: A branch is *complete* just in case every resolvable sentence in that branch has been resolved in that branch.

Tree Completion: A tree is *complete* if and only if every branch in the tree is complete.

Derivability

STL: $\Gamma \vdash \bot$ just in case there is a closed tree with root Γ.

Derivability: $\Gamma \vdash \varphi$ just in case $\Gamma, \neg\varphi \vdash \bot$.

Question 1: Why should we care about $\vdash$?

Answer: So far, we shouldn't, but soon we will show that: $\Gamma \vdash \varphi$ iff $\Gamma \vDash \varphi$.

Suppose: Let's suppose $\Gamma \vdash \varphi$ iff $\Gamma \vDash \varphi$ for now.

Question 2: How can we determine whether Γ is satisfiable?

Answer: Show that $\Gamma \nvdash \bot$.

Examples

1. Evaluate the following argument for validity:

   $\neg R \supset \neg Q$
   $P \wedge Q$
   $\therefore\ P \wedge R$

2. Show that $A \vee B, B \supset C, A \equiv C \vDash C$.
3. Show that $(P \supset Q) \equiv (\neg Q \supset \neg P)$ is a tautology.
4. Show that $A \equiv \neg A$ is a contradiction.
5. Show that $\{P, P \supset Q, Q \supset \neg P\}$ is unsatisfiable.
6. Show that $\{P \supset Q, \neg P \vee \neg Q, Q \supset P\}$ is satisfiable.
7. Evaluate $P, P \supset Q, \neg Q \vDash A$.