Notre Dame Journal of Formal Logic
Volume 28, Number 3, July 1987

More on Trees and Finite Satisfiability: The Taming of Terms

MIODRAG KAPETANOVIĆ and ALEKSANDAR KRAPEŽ

As Boolos showed in [1], the new rule proposed by Burgess provides (together with the usual tableaux rules) a simple and elegant method for testing finite satisfiability of first-order sentences. We present an extension of the method to the case of languages which contain function symbols. For these languages the usual rule "from $\forall x\phi(x)$ infer $\phi(t)$ for any term t" produces in some cases only infinite models. For instance, when applied to $\forall xRf(x)$ it gives us the infinite one-branch tree:

$$\begin{array}{c} \forall xRf(x) \\ | \\ Rf(a) \\ | \\ Rf(f(a)) \\ | \\ \vdots \end{array}$$

Where language contains equality we can in principle get rid of function symbols in standard way, replacing them by new predicate symbols (which are to serve as their "graphs"), but it is more natural to try to put up with terms.

In order to avoid ambiguities we first list some necessary definitions.

A *tableau* for a sentence ϕ is a tree built up by placing ϕ at the top node and then applying the following reduction rules.

($\neg\neg$) If $\neg\neg\phi$ lies on a branch B we extend it to (B, ϕ).

($\wedge$) If $\phi \wedge \psi$ lies on B, we extend it to (B, ϕ, ψ) (similarly for $\neg\vee$, $\neg\rightarrow$).

($\vee$) If $\phi \vee \psi$ lies on B, then B splits into two extensions (B, ϕ), (B, ψ) (similarly for $\neg\wedge$ and $\rightarrow$).

Received September 30, 1985; revised February 3, 1986