The tableau method for predicate logic

We present a simplified proof system for first order logic that will allow us to prove inconsistency, but not consistency.

More precisely, if A is a finite set of sentences, then

- if A inconsistent, i.e., does not have a model, we will get a proof of this fact
- if A has a model, the method gives no information.

The method generates a tableau that we will be able to close in case A is inconsistent, but that at any time we will not be able to say if it will close.

Theorem (Alonzo Church '36)

There is a computer program taking as inputs finite sets of sentences $\mathbb{A}$ s.t. the program terminates if and only if $\mathbb{A}$ is inconsistent.

On the other hand, there is no computer program that terminates if and only if $\mathbb{A}$ is consistent.

The derivation rules

(1) $$\begin{array}{c} \neg\neg B \\ | \\ B \end{array}$$

(2) $$\begin{array}{c} B \wedge C \\ | \\ B \\ C \end{array}$$

(3) $$\begin{array}{c} \neg(B \wedge C) \\ \diagup \qquad \diagdown \\ \neg B \qquad \neg C \end{array}$$

(4) $$\begin{array}{c} B \vee C \\ \diagup \qquad \diagdown \\ B \qquad C \end{array}$$

(5) $$\begin{array}{c} \neg(B \vee C) \\ | \\ \neg B \\ \neg C \end{array}$$

(6) $$\begin{array}{c} B \rightarrow C \\ \diagup \qquad \diagdown \\ \neg B \qquad C \end{array}$$

(7) $$\begin{array}{c} \neg(B \rightarrow C) \\ | \\ B \\ \neg C \end{array}$$

(8) $$\begin{array}{c} B \leftrightarrow C \\ \diagup \qquad \diagdown \\ \begin{array}{c} B \\ C \end{array} \qquad \begin{array}{c} \neg B \\ \neg C \end{array} \end{array}$$

(9) $$\begin{array}{c} \neg(B \leftrightarrow C) \\ \diagup \qquad \diagdown \\ \begin{array}{c} B \\ \neg C \end{array} \qquad \begin{array}{c} \neg B \\ C \end{array} \end{array}$$

(10)

$$\begin{array}{c} B \\ s = t \\ | \\ C \end{array}$$

where C is obtained from B by replacing one or more occurrences of the term s by the term t

(11)

$$\begin{array}{c} B \\ t = s \\ | \\ C \end{array}$$

—— " ——

(12)

$$\begin{array}{c} \forall x\, B \\ | \\ C \end{array}$$

where C is obtained from B by replacing each free occurrence of the variable x in B by some term t that occurs in a formula in the branch above "$\forall x B$"

(13)

$$\begin{array}{c} \exists x\, B \\ | \\ C \end{array}$$

where C is obtained from B by replacing each free occurrence of the variable x in B by a new constant term <u>not</u> occurring in any formula in the branch above "$\exists x B$".

(14) $\neg \forall x\, B$

$|$

$\exists x\, \neg B$

(15) $\neg \exists x\, B$

$|$

$\forall x\, \neg B$

<u>Note</u> We do <u>not</u> check off formulas that have been used since these can be reused at a later point. Also, by Church's theorem, we cannot conclude consistency simply by not being able to close the tableau.

We close a branch whenever we find a formula B and its negation $\neg B$ in the same branch. Similarly, if we find a formula " $\neg s = s$.

<u>Example</u> Show that

$$A = \{ \forall x\; ffx = x \;,\; \forall x\; fffx = x \;,\; \neg \forall x\; x = fx \}$$

is an inconsistent set of sentences by showing that its tableau closes.

$\forall x \; ffx = x$

$\forall x \; fffx = x$

$\neg \forall x \; x = fx$

$|$

$\exists x \; \neg x = fx$

$|$

(*) $\neg a = fa$ where a is a new constant symbol.

$ffa = a$

$fffa = a$

$|$

(**) $\neg a = fffa$ replacing the right hand occurrence of a in (*) by ffa

$|$

$\neg fffa = fffa$ replacing the left hand occurrence of a in (**) by ffa.

$\underline{\qquad\qquad}$

<u>Example</u> Show that the following argument is valid:

$$\frac{\exists x\,(Px \wedge \forall y\,(By \rightarrow Gxy))}{\forall y\,(By \rightarrow \exists x\,(Px \wedge Gxy))}$$

$$\exists x\,(Px \wedge \forall y\,(By \rightarrow Gxy))$$

$$\neg \forall y\,(By \rightarrow \exists x\,(Px \wedge Gxy))$$

new constant a : $Pa \wedge \forall y\,(By \rightarrow Gay)$

$$\exists y\, \neg (By \rightarrow \exists x\,(Px \wedge Gxy))$$

new constant b $\neg (Bb \rightarrow \exists x\,(Px \wedge Gxb))$

$$Bb$$

$$\neg \exists x\,(Px \wedge Gxb)$$

$$\forall x\, \neg (Px \wedge Gxb)$$

$$\neg (Pa \wedge Gab)$$

$$Pa$$

$$\forall y\,(By \rightarrow Gay)$$

$$Bb \rightarrow Gab$$

$\underline{\neg Bb}$ $\quad$ Gab

Gab

$\underline{\neg Pa}$ $\quad$ $\underline{\neg Gab}$