Errata and Addendum for Tableau Methods for Modal and Temporal Logics

August 24, 2000

page 323, Figure 7: Many of the systems marked with question marks in this table have been formulated successfully by Linh Anh Nguyen in:

```
@InProceedings{nguyen-tableau-superformula,
  author =       {Linh Anh Nguyen},
  title =        {Sequent-like Tableau Systems with the Analytic
                  Superformula Property for the Modal Logics
                  {KB}, {KDB}, {K5}, {KD5}},
  booktitle =    {TABLEAUX 2000: Automated Reasoning with Analytic
                  Tableaux and Related Methods},
  pages =        {341-351},
  year =         2000,
  editor =       {R Dyckhoff},
  volume =       {LNAI 1847},
  publisher =    {Springer}
}
```

Page 374, Figure 15: The systematic procedure contains a stupid mistake in the clause ($\vee$) for disjunctions. Replace the passage

old "If any of these updates fails to add the corresponding formula then delete that fork, possibly leaving $\mathcal{B}$ unaltered or with no fork" with

new "If any of these updates fails to add the corresponding formula then add a copy of that formula to the appropriate fork but mark that copy as asleep.

repurcussions Some of the subsequent proofs may have to be modified because they depend upon a branch not containing duplicates.

acknowledgement This error was pointed out to me on 12 November 1998 by Karl-Michael Schneider (`schneide@phil.uni-passau.de`) from the University of Passau, Germany. He also suggested the solution. More precisely he points out that

> It seems that there is a bug in the systematic labelled tableau construction given in fig. 15 on page 82. As an example, consider the set of formulas
>
> $$X = \{\neg P, \neg(P \wedge Q), \neg(\neg P \wedge \neg Q)\}$$

The set X is satisfiable in a propositional model where P is false and Q is true.

However, the systematic tableau construction in fig. 15 produces a closed tableau for X. The reason is that when the second formula $1 :: \neg(P \wedge Q)$ is processed, splitting of the branch fails because $1 :: \neg P$ is already on it, and so $1 :: \neg Q$ is added to the branch without producing a fork.

[stuff deleted]

Perhaps, when $1 :: \neg(P \wedge Q)$ is processed, a fork should be produced in any case, adding $1 :: \neg P$ to the branch and marking it finished if it is already on the branch.