$\boldsymbol{KE^+}$: Beyond Refutation

Guido Governatori

CIRFID, Università di Bologna, via Galliera 3, 40121 Bologna, Italy
E-mail: governat@cirfid.unibo.it

The system KE^+, a tableau-like proof system based on D'Agostino-Mondadori KE [DM94], is presented in this paper. This system avoids some of the drawbacks of other proof methods. In fact it is completly analytical, it is able to detect whether a formula is either a tautology or a contradiction or only a satisfiable one; in the course of a proof it can detect whether a subformula is a tautology and it uses this fact in the proof of the main formula.

In what follows we shall use the Smullyan uniform notation [Smu68]; if X is a signed formula, X^C denotes the conjugate of X.

The method KE^+ follows consists in verifying whether the truth of the conjugate of an immediate subformula of a β formula implies the truth of the other immediate subformula; if it is implied then we have enough information to affirm that the whole formula is provable. This result is obtained through the fact that in a given branch, the branch beginning with the conjugate, a formula which leads to the branch closure does not exist (i.e. there are not two formulas TA, FA) but this is done by proving that the conjugate of the formula occurs in the branch, i.e. we have to see that in a branch a signed formula appears twice, and that the two occurrences are derived from appropriate formulas.

KE and KE^+ share the same inference rules and differ only with respect to the proof procedure they use. The main feature of KE is that it is a method which uses elimination rules and an analytic form of cut (PB). Its rules are stated as follows:

$$\frac{\alpha}{\alpha_i}[\alpha\text{–rule}] \qquad \frac{\begin{array}{c}\beta\\ \beta_i^C\end{array}}{\beta_{3-i}}[\beta\text{–rule}] \qquad \frac{}{X \quad X^C}[PB] \qquad \frac{\begin{array}{c}X\\ X^C\end{array}}{\times}[PNC]$$

KE can be used either as a refutation method or as a direct method of proof, for more details about KE see [DM94]. Unfortunately, when KE is used directly, it has to check both the branch starting with the given formula and the branch starting with the conjugate of the given formula. KE^+ does not suffer this disadvantage, in fact it works straightforwardly with the formula to be proved.

Definition 1. *An α-formula is analysed in a branch when both α_1 and α_2 are in the branch; a β-formula is analysed in a branch when either: if β_1^C is in the branch also β_2 is in the branch, or if β_2^C is in the branch also β_1 is the branch. A β formula will be called fulfilled in a branch if: either β_1 or β_2 depending on β occurs in the branch, or either β_1 or β_2 is obtained from applying PB on β.*

Each formula depends on itself; a formula B depends on A either if it is obtained through an application of the α-rule or it is obtained through an application of KE's rules on formulas depending on A; a formula C depends on A, B

if it is obtained through an application of a β-rule where A, B are its premises; the formulas obtained through PB depend on the formula PB is applied to; if C depends on A, B then C depends on A and C depends on B.

Definition 2. *A branch is E-completed if all the formulas occurring in it are analysed; a branch is completed if it is E-completed and all the β-formulas occurring in it are fulfilled. We shall call a branch a β^C-branch if its root is obtained applying PB on a β-formula and it starts with β_i^C; and each branch generated by PB on a formula occurring in a β^C-branch is a β^C-branch. Any branch which is not a β^C-branch and is obtained from PB will be called a β-branch. We shall call a branch a $\top$-branch if it contains only formulas which are equivalent to $\top$ and the formulas depending on them.*

The procedure starts from the formula to be proved, then (1) it selects a β^C-branch ϕ which is not yet completed and which is the β^C-branch with respect to the greater number of formulas; (2) if ϕ is not E-completed, it expands ϕ by means of the α- and β-rules until it becomes E-completed; (3) if the resulting branch is neither completed nor closed then it selects aformula of type β which is not yet fulfilled in the branch, if possible a β-formula which results from step 2, and then it applyes PB with β_1, β_1^C (or, equivalently β_2, β_2^C) then it applyes step 1; otherwise it returns to step 1

Theorem 1. *A formula $A \equiv \top$ if either: (1) in one of the β^C-branches it generates there is a formula which appears twice, and one occurrence depends on $\beta_i^C, i \in \{1, 2\}$ and the other depends on β, or (2) each β^C-branch is closed and the β-branches contain $\top$, or (3) each β^C-branch is a $\top$-branch.*

Preliminary research into KE^+'s complexity and efficiency shows that for certain classes of formulas it is more efficient than KE. For example, given the tautology $\alpha \to (\beta \to \alpha) \equiv ((\neg\alpha \vee \beta) \equiv (\alpha \to \beta))$, its shorter and longer proofs, using KE, consists respectively of 24 and 36 (34) steps, whereas the analogous proofs using KE^+ spend respectively 10 and 19 steps; on the other hand, if we query the systems with the negation of the above tautology both trees are 23 steps long, but KE^+ tells us that the formula is a contradiction, whereas all information that KE gives us is that the negation of the formula is satisfiable, but we are not able to know whether the formula itself is satisfiable.

The approach we have presented can work side by side with KE and it is useful to build more efficient theorem provers, because, with a pre-analysis of the formula, we can choose the best strategy to follow in order to prove it, i.e. we can choose, according to its structure, whether to refute it is more economical than proving it directly, i.e. which system, for a given formula, is more efficient.

References

[DM94] Marcello D'Agostino and Marco Mondadori. The Taming of the Cut. *Journal of Logic and Computation*, 4, 1994: 285–319.

[Smu68] Raymond Smullyan. *First-Order Logic.* Springer-Verlag, Berlin, 1968.