A Method to Solve the Puzzles of Knights and Knaves [1]

László Aszalós

In his book "What is the title of this book?" Raymond M. Smullyan used a lot of puzzles to illustrate the background of the Gödel incompleteness theorem. These puzzles became popular and nowadays are being published in amusement magazines too. In each section of the book different conditions are met. In the best known type of puzzles we have only two types of people, knights and knaves. Knights always tell the truth and knaves always lie.

Since in each puzzle there are only finitely many characters, we shall work with a finite number of characters, too. For the sake of simplicity, we denote them by a, b, c, ... To formalize the puzzles we need to extend the concept of a formula. In addition to usual connectives we shall use the symbols T_x, F_x, S_xA and C_xA. Here x denotes somebody among a, b, c, ... and A denotes a formula that may contain S_x or C_x; hence, we allow the nesting of these modal operators with no restriction, but the formula must be finite, of course. The reading of T_x, F_x, S_xA and C_xA are: x is a knight, x is a knave, x said that A and x can say that A, respectively. To construct a model we need to know about each person whether is he (she) a knight or a knave, and what he (she) said. If x is a knight, then T_x will be true and F_x will be false, and if x is a knave, then the opposite will be the case. If x said that A, then S_xA will be true; otherwise, it will be false.

In Smullyan book the characters are polite, they answer all the questions. If we take the general case, we can no more expect this. If somebody said n sentences, he does not need to say a new one, even if he could. A knight could say all the true formulae, which are infinitely many, but he always said a finite number of them. For this reason we shall work with *could say* or *can say*, with the modal operator C_x. The formula C_xA will be true if A is true and x is a knight or A is false and x is a knave. In all other cases C_xA will be false.

The are several methods for solving the puzzles. Smullyan's method is based on the type of characters. However, later, when we will have a lot of cases, this method will be uncomfortable and inefficient. David Gries rewrote the formulas using the logical law $S_xA \to (A \equiv T_x)$ and afterwards he was able to work with formulae of propositional logic. This rewriting was based on the fact that $Tx \equiv \neg Fx$. In treating the subsequent sections of the book we shall loose this nice property. Larry Wos rewrote the puzzles as first order formulae and used the theorem prover Otter to solve them. Using this method we can solve complicated puzzles, too, but there is no complete algorithm for solving all the first order formulae, which is due to the incompleteness theorem.

For this reasons I constructed my own method to solve this and other puzzles of the book. This method is based on the well-known method of analytic tableaux. I needed only to add new rules for the new type of formulae. I show that my method can be extended to the puzzles formulated with a third type of people - normal people - who can say anything. At the end of this article I describe the heart of my prover that can solve all these puzzles.

[1]Research was partly supported by the grant OTKA no. 354-19341