ABSTRACT

A problem solver is a common tool used in solving Artificial Intelligence problems. They compute solutions based on the their knowledge about the current status of the world they operate in by applying a set of operators or rules[1]. *The current status of the world (facts) is stored in the form of a knowledge base which is basically expert knowledge in a certain domain. For an expert in the field, contributing the solution may not be satisfactory if the system does not give the rationale behind conclusion. For example a medical system that suggests that a patient's leg be amputated without providing a valid justification for the conclusion may not be acceptable. Hence, the need for truth maintenance systems. Apart from providing a valid justification for the conclusion, a truth maintenance system supports dependency driven backtracking, supports default reasoning and recognizes inconsistencies when they arise in a system.*

The thesis of this dissertation is to develop an integrated program that solves a wide variety of differentiation and integration problems by the application of rules of calculus. If a solution exists, explanations as to which rules were applied to solve the problem are displayed on the screen as solution process is executed. The user is duly made aware of any unsuccessful attempt to solve a problem. We extend JSAINT, which is a Justification-based Truth Maintenance System version of its predecessor, SAINT. SAINT was of the first Artificial Intelligence (AI) programs developed at Massachusetts Institute of Technology (MIT) in 1963 and was used to compute a vast range of indefinite integration problems that appeared on an MIT calculus exam.

[1]Procedural knowledge is expressed in rules and like assertions, are stored in a database. Each rule has two parts, a *trigger* and a *body*. The trigger defines the patterns which the rule responds to and the body is the code to be executed when an assertion unifies with the trigger.